Plattformunabhängiger Austausch von Bilddaten für die Patientenkommunikation

Fakultät für Medien und Informationswesen
der Hochschule Offenburg

Bachelor Thesis

zur Erlangung des akademischen Grades
Bachelor of Science

vorgelegt von

Pascal Ilg
geboren am 16.11.1990 in Pforzheim

Unternehmens- und IT-Sicherheit
Sommersemester 2015

Erstprüfer:	Prof. Dr. rer. nat. Tom Rüdebusch
Zweitprüfer:	Steffen Hennhöfer
Firmenadresse:	Sirona Dental Systems GmbH Fabrikstraße 31 D-64625 Bensheim Germany

Erklärung

Hiermit versichere ich, die vorliegende Abschlussarbeit selbstständig und nur unter Verwendung der von mir angegebenen Quellen und Hilfsmittel verfasst zu haben. Sowohl inhaltlich als auch wörtlich entnommene Inhalte wurden als solche kenntlich gemacht. Die Arbeit hat in dieser oder vergleichbarer Form noch keinem anderem Prüfungsgremium vorgelegen.

Bensheim, 18.08.2015

Pascal Ilg

Abstract

Diese Thesis befasst sich mit einer Erweiterung für die bestehende Software SIDEXIS 4 der Firma Sirona GmbH. Die Funktion der Erweiterung besteht darin, Röntgenbilder eines Patienten zum Herunterladen bereit zu stellen. Dabei werden die Bilddateien verschlüsselt auf eine Online-Plattform hochgeladen und ein Quick-Response-Code (QR-Code) zu der Internet-Adresse der Ressource generiert. Der Patient hat die Möglichkeit, durch Abfotografieren des QR-Codes, seine Röntgenbilder einzusehen und diese zu speichern. Fragen in Bezug auf die Datensicherheit im Bereich von Patientendaten, werden ebenfalls ausführlich beschrieben und behandelt.

Die Motivation zur Entwicklung dieser Erweiterung ist eine flexible, moderne Patientenkommunikation. Dadurch kann der Patient seine Bilder digital mit sich führen und gegebenenfalls weiter verwalten.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

AES	Advanced Encryption Standard
App	Application software program
ASP.NET	Active Sever Pages .NET
BDSG	Bundesdatenschutzgesetz
BGB	Bürgerliches Gesetzbuch
BSI	Bundesamt für Sicherheit in der Informationstechnik
CLR	Common Language Runtime
CPU	Central Processing Unit - Prozessor
EC2	Elastic Compute Cloud
EU	Europäische Union
GB	GigaByte
GmbH	Gesellschaft mit beschränkter Haftung
HTTPS	Hypertext Transfer Protocol Secure
IEC	International Electrotechnical Comission
IaaS	Internet as a Service
IOS	internet operating system
ISMS	Informationssicherheits-Managementsystem
ISO	International Organization for Standardization
JPEG	Joint Photographics expert Group
KB	KiloByte
MB	MegaByte
MVC	Model-View-Controller
NIST	National Institute of Standards and Technology
OHG	Offene Handelsgesellschaft
PaaS	Plattform as a Service
PHP	Hypertext Preprocessor
QR-Code	Quick-Response-Code
RAM	Random-Access Memory - Arbeitsspeicher
RöV	Röntgenverordnung

SaaS	Software as a Service
SDK	Software Development Kit
SSH	Secure Shell
STGB	Strafgesetzbuch
TB	TerraByte
TLS	Transport Layer Security
URL	Uniform Resource Locator
UTC	Coordinated Universal Time
VM	Virtuelle Maschine
VPN	Virtual Private Network
WPF	Windows Presentation Foundation

1 Einleitung

1.1 Problemstellung

Die Bereitstellung von Röntgenbilder ist für Ärzte und Patienten mit einem höheren Aufwand verbunden. Wenn Röntgenbilder einem Patienten ausgehändigt werden sollen, werden in der Regel 2 Varianten dazu genutzt. Da sich die Röntgenbilder lediglich lokal auf einer Festplatte der Arztpraxis befinden, können diese ausgedruckt werden. Dies erfordert einen speziellen Drucker. Für eine hochwertiges Bild wird zudem ein kostspieliger Röntgenfilm benötigt. Hat der Patient eine Ausdruck erhalten, muss er diesen sicher verwahren um sicherzustellen das er nicht beschädigt wird oder verloren geht. Eine weitere Möglichkeit zur Bereitstellung wird durch Brennen der Bilder auf eine CD realisiert. Da die Bilder in einem hohen Format vorliegen, werden dafür meist DVDs mit höherer Speicherkapazität benötigt. Der Brennvorgang nimmt dafür einen erheblichen Zeitraum in Anspruch. Das gebrannte Medium muss ebenfalls sicher verwahrt werden und bietet Anfälligkeit für Beschädigung. Nicht zuletzt kann dem Patienten für die Bereitstellung der Bilder ein entsprechendes Entgelt erhoben werden um die Kosten des Drucks, oder der CD/DVDs auszugleichen. [45]

Das Bereitstellen der Bilder soll für Ärzte und Patienten bequemer, moderner, günstiger und vor allem unkompliziert gestaltet werden. Die Idee umfasst dabei das Bereitstellen der Röntgenbilder mithilfe einer Online-Plattform. Über diese soll der Patient seine Röntgenbilder in einem plattformunabhängigen Format Herunterladen können. Dadurch würde der Patient seine Röntgenbilder in digitaler Form erhalten und kann diese weiter verwalten. Da mittlerweile jedes Smartphone standardmäßig mit einer Kamera ausgestattet ist und laut einer Statistik von statista[1] die Nutzung von Smartphones in Deutschland immer stärker zunimmt, bietet sich die Verwendung eines QR-Codes zum Verweisen auf die Bilder in der Cloud an. Durch die umstandslose Funktion der Kamera erhält der Patient in einem Schritt, durch einfaches abfotografieren, Zugriff zu seinen Bildern.
Eine Alternative Übertragungen wie über Bluetooth wäre ebenfalls möglich. Im Gegensatz zum QR-Code ist diese Übertragung allerdings mit mehreren Schritten durch den Patienten verbunden. Diese bestehen aus aktivieren der Bluetooth Funktion, suchen der korrekten Bluetooth-Quelle und dem Verbinden durch Pairing. [22]

Diese Thesis bezieht sich auf die Auswahl der Plattform zur Bereitstellung der Röntgenbilder und der Implementierung eines Prototypen. Es muss evaluiert werden, welche Plattform dafür am besten geeignet ist und welche Anforderungen erfüllt sein müssen. Zudem müssen

[1] statista ist ein Statistik Portal mit Statistiken und Studien aus über 18.000 Quellen

im Bezug auf den Datenschutz die Richtlinien zum Auslagern von Patientendaten überprüft werden, ob eine Auslagerung zulässig ist und welche Sicherheitsvorkehrungen dafür notwendig sind.

1.2 Zielsetzung

Im Rahmen dieser Bachelor Thesis soll für die Firma SIRONA ein Konzept entworfen werden, das eine Kommunikation mit dem Patienten zum Bereitstellen seiner Röntgenbildern ermöglicht. Dabei soll ein Modul, ähnlich einem Plugin, für die bestehende Software SIDEXIS4 [23], welche zum Aufnehmen, Organisieren und Diagnostizieren von Röntgenbildern dient, und in Zahnarztpraxen zum Einsatz kommt, implementiert werden.(weitere Informationen zu SIDEXIS 4 sind in Kapitel 3, Grundlagen, näher beschrieben.)

Mithilfe des Moduls, können Röntgenaufnahmen auf eine Plattform hochgeladen werden, wobei die dem Datenschutz entsprechenden Sicherheitsaspekte erfüllt sind. Dazu gehören vor allem eine verschlüsselte Übertragung und das Entfernen signifikanter Merkmale, wie dem Name des Patienten, sodass keine Rückschlüsse auf eine Person möglich sind.

Anhand eines generierten QR-Codes, soll es dem Patienten möglich sein, die Röntgenbilder durch abfotografieren, mithilfe eines Smartphones und einer QR-Reader-Application software program (App)[2], herunter zu laden. Der QR-Code soll über eine Uniform Resource Locator (URL) auf die Ressource verweisen. Die hoch geladenen Bilder sollen eine bestimmte Zeit zum Herunterladen zur Verfügung stehen, bevor diese dann wieder entfernt werden.

Dazu wurde ein Pflichtenheft[3] erstellt, das die Anforderungen von Patienten, sowie die Vorgaben und Kriterien beschreibt.

1.3 Gliederung und Vorgehensweise

Die Vorgehensweise dieses Projektes umfasst folgende Phasen:

1.3.1 Vorbereitungsphase

Das Kapitel 1, Einleitung, gibt einen Einstieg in die Thematik der Arbeit und fasst die Problemstellung, sowie die Zielsetzung zusammen. Die Einleitung umfasst ebenfalls die Vorgehensweise und Gliederung bei der Durchführung. Die Recherche vorhandener Systeme wird in Kapitel 2, Verwandte Arbeiten, beschrieben.

[2]Software auf dem Mobiltelefon zum lesen und umwandeln eines QR-Codes.
[3]Anlagen: Das Pflichtenheft beschreibt die Anforderungen für die Umsetzung der Implementierung

Nach Abschluss der Recherche sollte das Basiswissen als Voraussetzung für spätere Konzepte vorhanden sein. Wichtiges ist daher im Kapitel 3, Grundlagen, erklärt. Das Kapitel 4, beinhaltet das Konzept. Dieses beschreibt das Ziel, Einschränkungen zum Datenschutz, geltende Anforderungen, Anwendungsansätze der Cloud-Systeme und deren Evaluation, eine Gesamtbetrachtung der Evaluierten Systeme mit anschließender Entwurfsentscheidung und der Beschreibung des Systems unterteilt in verschiedenen Komponenten.

Anhand dieser Informationen wird ein Implementierungskonzept erstellt, an welchem die weiteren Phasen anknüpfen.

1.3.2 Durchführungsphase

Die Durchführungsphase beschäftigt sich mit der Umsetzung des Konzeptes und der Dokumentation. Programmiertechnisch werden die Anforderungen dem Konzept zufolge implementiert. Diese Phase wird im Kapitel 5, Implementierung behandelt.

1.3.3 Abschlussphase

In der letzten Phase, wird das Projekt auf die Funktionalität getestet und bei auftretenden Fehlern korrigiert, bevor es dann an zuständiger Stelle freigegeben werden kann. Außerdem beschreiben Ausblick und Fazit in Kapitel 6, Perspektiven über zukünftige Weiterentwicklungen und Ansätze. Letztendlich folgt eine abschließende persönliche Meinung zu der Arbeit.

2 Verwandte Arbeiten

In diesem Kapitel werden bereits existierende Arbeiten die dem der Thesis ähneln vorgestellt. Das Hauptaugenmerk liegt besonders bei solchen verwandten Arbeiten, bei denen ebenfalls eine Cloud oder der QR-Code zum Einsatz kommt. Diese sind nicht identisch in ihrer Funktionalität und dem Verwendungszweck, sondern geben vielmehr einen Einblick auf die unterschiedlichen Möglichkeiten der Nutzung.

2.1 HealthDataSpace

Inhalte des folgenden Abschnittes sind der Produktwebseite von HealthDataSpace entnommen [11].

Eine bereits existierendes Anwendung zur Patientenkommunikation bietet HealthDataSpace. Dies umschließt ein Netzwerk für Patienten und Ärzte, zum Verwalten und Befunden von digitalen Bildern. Diese Anwendung wird in Zusammenarbeit der beiden Unternehmen Telepaxx und Digithurst bereitgestellt.
Telepaxx entstand im Jahre 1996 und und ist marktführend in Europa im Bereich der Datenschutz-Zertifizierten-Langzeitarchivierung von Medizindaten.
Digithurst wurde 1983 gegründet und hat als bekanntes IT-Unternehmen für Radiologie und Bildbetrachtungswerkzeuge die Software für HealthDataSpace entwickelt.

Das Konzept von HealtDataSpace zielt darauf ab, Patientendaten, die normalerweise in verschiedenen Einrichtungen auf diversen Datenspeichern abgelegt werden, zentral und patientenzentriert bereitzustellen. Der Patient bekommt die Möglichkeit, sich über HealthDataSpace mit Ärzten und Kliniken zu vernetzen. Er kann entscheiden, mit welchem Arzt er seine Dokumente teilen will. Für Ärzte können die Patientendaten automatisch in der HealthDataSpace abgelegt werden. Der Arzt kann ebenfalls datenschutzkonform, Medizindaten mit Kollegen, Spezialisten und Kliniken teilen.

Die Nutzung ist einfach und benötigt keine zusätzliche Software oder andere Installationen, sondern ist jederzeit über den Webbrowser verfügbar. HealtDataSpace ist auf verschiedenen Geräten plattformunabhängig nutzbar. Dazu gehören Smartphones, Laptops oder Tablets. Bezüglich der verwendeten Technologie finden sich keine weiteren detaillierteren Informationen.

2.2 Planmeca Romexis

Inhalte des folgenden Abschnittes sind der Produktwebseite von Planmeca entnommen [44].

Die Planmeca GmbH existiert seit 1971 und entwickelt in der Dental Branche vergleichbare Produkte wie sie auch SIRONA anbietet. Neben deren Software Romexis, zur Bildbearbeitung für Rechner und Mobilgeräte, bieten sie auch eine Cloud als Bildübertragungsservice an.

Die Nutzung wird allerdings nur kostenpflichtig offeriert. Planmeca stellt dazu drei unterschiedliche Varianten zur Verfügung, wobei der Unterschied von der Menge des monatlichen Datentransfers abhängt. Diese Angebote sind hauptsächlich für Ärzte und Kliniken gedacht, welche ihren Patienten einen Austausch von Bilddaten ermöglichen wollen. Entscheidet sich eine Praxis, oder Klinik für eine der Varianten, so können diese Röntgenbilder für ihre Patienten oder andere Praxen freigeben.
Mit dem kostenlosen Planmeca Romexis Viewer können gebührenfrei Bilder empfangen und angesehen werden. Hierfür fallen keine weiteren Kosten an.

Um den Schutz der Daten gewährleisten zu können, werden alle Daten und Übertragungen mit einerm 256-Bit-Algorithmus verschlüsselt.

2.3 Kunst im öffentlichen Raum Frankfurt

Inhalte des folgenden Abschnittes sind der Projektwebseite von "Kunst im öffentlichen Raum Frankfurt" entnommen [29].

Mit dem Projekt "Kunst im öffentlichen Raum Frankfurt", hatte das Kulturamt Frankfurt am Main die Idee, QR-Code's auf Edelstahltafeln, an Denkmälern, Brunnen und anderen Kunstwerken der Stadt Frankfurt anzubringen. Mit dem QR-Code als "*Schnittstelle zur digitalen Welt*" [29], sollen so mehr Informationen bereitgestellt werden. Dabei verweisen diese auf Inhalte des bereits bestehenden Portals.
Der Gebrauch ist für den Anwender sehr umstandslos gestaltet, was auch Ziel und Anforderung für diese Bachelorarbeit ist. Somit gelangt man durch Abfotografieren der Schilder mit dem Smartphone und einem QR-Code-Reader[1] auf eine mobile Webseite mit ausführlichen Informationen des Künstlers, über Entstehung, Aufstellung und Erzählung zu dem Werk. Dieses Projekt ermöglicht den Betreuern detailliertere und größere Mengen von Informationen auf einer Webseite zu platzieren, anstatt diese auf Informationstafeln vor Ort anzubringen. Aktualisierungen von Informationen können jederzeit digital vorgenommen werden, ohne das neue Informationstafeln zur Ausstellung erstellt werden müssen.

So stellen die bis jetzt angebrachten QR-Codes im Raum Frankfurt, bei einem zweistün-

[1] QR-Code-Reader: Software für das Mobiltelephon zum Lesen und Decodieren des Codes

digen Rundgang, Informationen über die Stadtgeschichte bereit. Mithilfe eines erstellten Stadtplans können auch auswärtige Besucher die Objekte finden, welche auf dem Stadtplan entsprechend markiert sind.

2.4 Vinopass

Inhalte des folgenden Abschnittes sind der Projektwebseite von Vinopass und der Produktwebseite der Werbeagentur entnommen [19] [18].

Vinopass ist eine Erfindung der 2001 gegründeten Offene Handelsgesellschaft (OHG) Medienagenten, welche Dienstleistungen als Full-Service-Werbeagentur leistet. Als ein Service von Medienagenten bietet Vinopass eine mobile Weinvermarktung mittels QR-Code an. Dieser wird dazu zusätzlich zum Etikett, auf der Flasche angebracht. Beim Abfotografieren können somit alle Informationen zu dem Wein auf einer Online-Webseite bereitgestellt, sowie Anbaugebiete bildlich veranschaulicht werden, um dem Kunden das Produkt näher zu bringen.

Vinopass stellt die QR-Codes und die Webseiten mit den Informationen zu den Weinen bereit. Mithilfe von Social Media Anbindungen, wie Facebook oder Twitter, können diese geteilt und weiterverbreitet werden. Durch diese Methode der automatischen Marketing Strategie, wird die Werbung der Produkte von den Kunden selbst realisiert. Außerdem erfasst Vinopass eine Nutzerstatistik der Seitenaufrufe und ermöglicht eine flexible Datenhaltung und Modifizierung.
Es werden Vinopässe verkauft, wobei ein Pass einer Weinsorte entspricht. Diese Vinopässe können als Abo monatlich gezahlt werden. Insgesamt gibt es sechs verschiedene Angebote mit unterschiedlicher Anzahl an Pässen.

Vinopass ist ein Beispiel für erfolgreiches Integrieren von QR-Codes im Bereich der Lebensmittel Vermarktung.

2.5 WhatsApp - Web

Inhalte des folgenden Kapitels sind der Projektwebseite von WhatsApp entnommen [6].

WhatsApp ist ein Dienst für Multimediageräte und wird hauptsächlich auf Smartphones und Tablets verwendet. Dieser ermöglicht die Kommunikation durch Textnachrichten. Zudem ist das Senden von Bildern, Videos, Audiodateien, Kontaktdaten, oder dem Standort möglich. Um den dienst auch mit dem Computer über den Browser verwenden zu können, wird die Authentifizierung mithilfe eines QR-Codes hergestellt.

Dazu muss der Nutzer über den Computer mithilfe eines Browsers folgende Webseite aufrufen:

www.web.whatsapp.com

Auf dieser Webseite, wie in Abbildung 2.1 dargestellt, wird ein QR-Code generiert und dem Nutzer angezeigt. Der QR-Code beinhaltet einen uniquen Identifikator. Der Nutzer hat die Möglichkeit über den installierten Whatsapp-Dienst auf seinem Smartphone den QR-Code abzufotografieren. Wurde der QR-Code abfotografiert, verwendet die App den Identifikator zusammen mit einem eindeutigen Zugangsschlüssel des Nutzers und sendet diese an den Whatsapp-WebServer. Der Webserver kann dadurch referenzieren, welchen Chat er für den Whatsapp-Nutzer in dem Browser darstellen muss. Dadurch wird dem Nutzer das Verfassen und Zugreifen seiner Nachrichten ermöglicht.

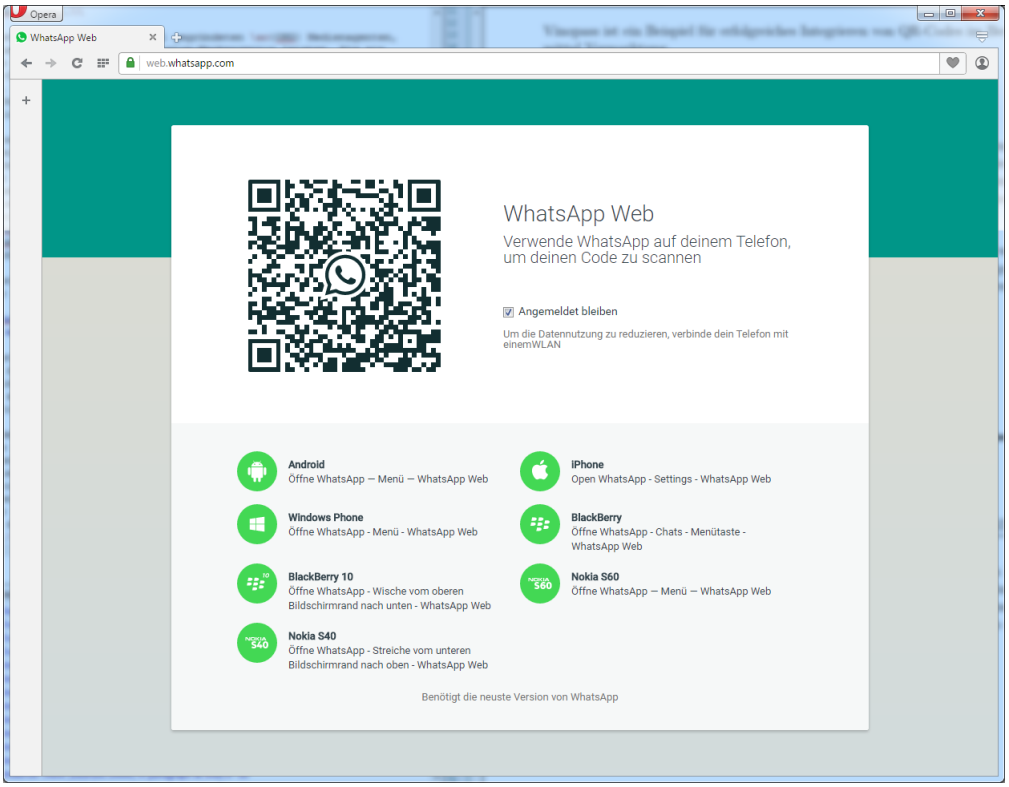


Abbildung 2.1: Whatsapp - Web Authentifizierung [6]

3 Grundlagen

Im Kapitel der Grundlagen wird auf verschiedene Systeme, Software und Algorithmen eingegangen. Diese sind als Grundlage dieser Thesis von Bedeutung und finden in späteren Kapiteln Verwendung (siehe dazu Kapitel 4, Konzept und Kapitel 5, Implementierung).

3.1 SIDEXIS 4

SIDEXIS 4 ist eine Software der Firma Sirona und bietet Funktionen zur Akquise, Verwaltung, Analyse, Befundung, Präsentation und Weitergabe von digitalen Bilddaten, wie Röntgenbilder für medizinische Zwecke im Bereich der Zahnheilkunde. Die Zielgruppe betrifft Zahn- und Fachärzte der Medizin, sowie deren Fachpersonal. [46]

3.1.1 Aufbau der Software

Die Benutzeroberfläche wird in 5 Phasen unterteilt. Diese werden nachfolgend grob beschrieben:

- **Start**: In dieser Phase wird eine Auftrags- und eine Terminliste dargestellt. Diese können dort angelegt werden, oder von einer Praxis-Verwaltungs-Software an SIDEXIS 4 gesendet werden.
- **Patient**: Hier werden alle geführten Patienten der SIDEXIS 4 Datenbank aufgelistet und es können neue Patienten angelegt werden. Ein Patient kann angemeldet werden, um alle bestehenden Röntgenbilder einsehen zu können.
- **Aufnahme**: In dieser Phase können Röntgenaufnahmen, von im Netzwerk vorhandenen Geräten und Sensoren, getätigt werden. Dazu können zur Indikation verschiedene Angaben eingegeben werden.
- **Untersuchung**: Bei der Untersuchung findet die Analyse und Befundung der Röntgenbilder statt. Es werden ebenfalls umfangreiche Werkzeuge für 2D-Bilder und 3D-Volumen zur Verfügung gestellt.
- **Ausgabe**: Die Phase der Ausgabe ermöglicht das exportieren der Bilddaten, wobei das Format[1] des Exports in den SIDEXIS 4 Einstellungen festgelegt werden kann. Der

[1] Gültige Formate sind JPG, PNG, BMP und TIFF

Druck der Bilddaten sowie das versenden via E-Mail, findet ebenfalls in der Ausgabe statt.

[46]

3.1.2 Framework

SIDEXIS 4 wurde mit der Entwicklungsumgebung Visual-Studio in der Hochschsprache C# und mit dem .NET Framework 4.5 programmiert. Dies ist wichtig für weitere Planungen der späteren Implementierung des Moduls und für die Auslagerung der Bilddaten zur Patientenkommunikation, welche an der Software anknüpfen soll.

3.2 QR-Code

In diesem Kapitel wird der QR-Code genauer beschrieben, wie dieser aufgebaut ist und benutzt werden kann. Der QR-Code dient für das Programm worüber diese Thesis handelt als Vermittler zwischen dem Patienten und den Bilddaten.

Der QR-Code ist eine Erfindung der japanischen Firma Denso Wave aus dem Jahr 1994. Dieser ist zweidimensional und kann Informationen in Form einer Zeichenfolge speichern. Es gibt bis zu 40 QR-Code-Versionen. Je nach Version beinhaltet der QR-Code eine unterschiedliche Menge von Modulen. Ein Modul besteht aus einem weißen oder schwarzen Punkt. Die größte Version ”40” mit 177 x 177 Modulen fasst bis zu 7089 Ziffern, oder 4296 alphanumerische Zeichen im Gegensatz zu dem konventionellen Barcode, bei dem nur bis zu 20 Ziffern gespeichert werden können. [17]

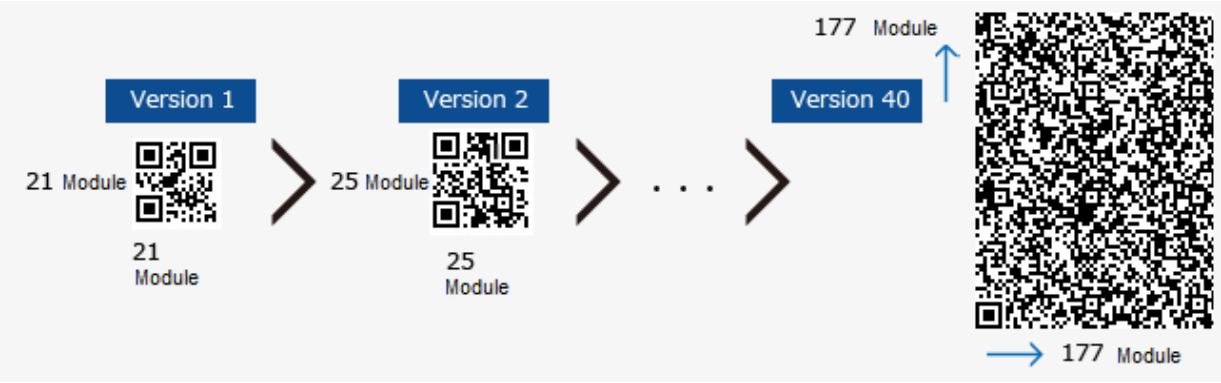


Abbildung 3.1: QR-Code Module [17], bearbeitet

Zudem ist der QR-Code resistent gegen Beschädigung oder Unkenntlichkeit durch Schmutz. Es gibt 4 verschiedene Level zur Fehlerkorrektur, die mithilfe des Reed-Solomon-Code[2] rea-

[2]Algorithmus zur Erkennung und zum korrigieren von Übertragungs- und Speicherfehlern. Im vgl. dazu W.W. Peterson and E.J. Weldon, Jr., Error-Correcting Codes, 2nd edition, MIT Press: Cambridge, Mass., 1972.

lisiert wird. Das heißt, bei Version 40 und der Fehlerkorrektur Level H können ca. 30% des Codes beschädigt oder unkenntlich sein und die Informationen können noch immer erkannt werden wie in Abbildung 3.2 zu erkennen ist.[17]

Version	Modules	ECC Level	Data bits (mixed)	Numeric	Alphanumeric	Binary	Kanji
40	177x177	L	23,648	7,089	4,296	2,953	1,817
		M	18,672	5,596	3,391	2,331	1,435
		Q	13,328	3,993	2,420	1,663	1,024
		H	10,208	3,057	1,852	1,273	784

Abbildung 3.2: QR-Code Versionen [17]

3.2.1 Struktur eines QR-Codes

Die Struktur eines QR-Codes beinhaltet unterschiedliche Bereiche, die für verschiedene Funktionen reserviert sind:

Die Positionserkennung, siehe (Abbildung 3.3, Markierung **1**), befindet sich in 3 Ecken und hilft dem Algorithmus der Software sich an der Bilddatei zu orientieren, um die relevanten Bereiche des Codebildes zu erfassen.
Um die Positionserkennung herum sind jeweils freie Zonen, siehe (Abbildung 3.3, Markierung **2**), als Abgrenzung an weitere Bereiche vorgesehen.
Der Takt, auch Synchronisation genannt, siehe (Abbildung 3.3, Markierung **3**), dient zur Erkennung der Größe des QR-Codes durch abwechselnde schwarz-weiß Punkten.
Das Element zur Lage-Erkennung, siehe (Abbildung 3.3, Markierung **4**), ermöglicht die Lesbarkeit, sodass der QR-Code aus jeder Rotation und jedem Winkel abfotografiert werden kann.
Der Bereich des Formats, siehe (Abbildung 3.3, Markierung **5**), welcher an die weißen freien Zonen der Positionserkennung anschließt, gibt Auskunft über das Level der Fehlerkorrektur und der Maske mit welcher die Nutzerdaten konvertiert wurden.
Die eigentlichen Nutzerdaten werden in einen ”bit-Stream” konvertiert und in 8-Bit-Blöcken im Nutzerdatenbereich, siehe (Abbildung 3.3, Markierung **6**), gespeichert.
Die Daten für die Fehlerkorrektur, siehe (Abbildung 3.3, Markierung **7**), werden ebenfalls wie im Nutzerdatenbereich in 8-Bit-Blöcken in dem vorgesehenen Bereich gespeichert.
Können Daten- und Fehlerkorrektur-Bits nicht in 8-Bit-Blöcke aufgeteilt werden, wird der Rest in einem Bereich für restliche Bits, siehe (Abbildung 3.3, Markierung **8**), gespeichert. [39] [43]

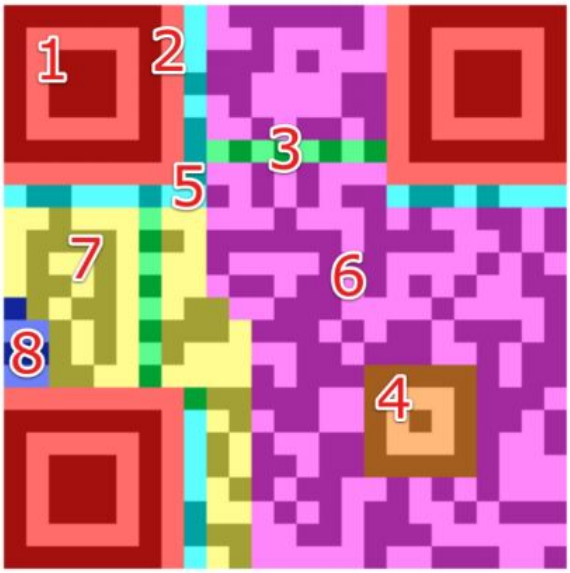


Abbildung 3.3: Struktur eines QR-Codes [43]

3.2.2 Verwendung

Um einen QR-Code verwenden zu können, wird eine Software als App benötigt und ein Smartphone oder Tablet, um den Code abfotografieren und entschlüsseln zu können. QR-Codes werden zu verschiedenen Zwecken verwendet. Zum Hinterlegen von Kontaktdaten, weiteren Informationen zu Produkten, oder auch zum einfachen Verbreiten von Webadressen. Dabei erkennt eine QR-Reader-App im Normalfall eine URL und leitet den Benutzer direkt an die Webseite weiter. Dazu finden sich Referenzprojekte im Kapitel 2, Verwandte Arbeiten. In der nachfolgenden Abbildung 3.4 wird der Ablauf der Verwendung eines QR-Codes dargestellt. Demnach wird der projizierte QR-Code mit einem Smartphone abfotografiert, wobei der QR-Reader die Daten deokodiert und anschließend darstellt. [7]

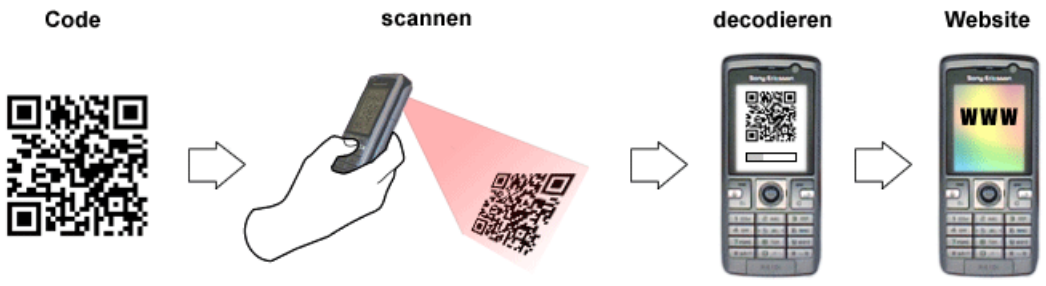


Abbildung 3.4: Ablauf der Verwendung eines QR-Codes [7]

Der QR-Code erfüllt die Anforderung [1.2] dieser Thesis, dessen Nutzung für den Patienten einen leichten und bequemen Bedienungsablauf gewährleistet. Statt eine Telefonnummer,

oder Web-Adresse abzutippen, kann durch bequemes abfotografieren eine Zeichenfolge übermittelt werden.

3.3 Cloud-Computing

In diesem Kapitel geht es um die Bedeutung von Cloud-Computing und die verschiedenen Formen und Auslegungen die eine Cloud besitzt. Da die Cloud in dieser Thesis ein Hauptfaktor darstellt, sind diese Informationen für spätere Konzeptentwürfe, sowie Anforderungen und die Systemauswahl wichtig.

Cloud-Computing wird vom Bundesamt für Sicherheit in der Informationstechnik (BSI) wie nachfolgend definiert:

> *"Cloud Computing bezeichnet das dynamisch an den Bedarf angepasste Anbieten, Nutzen und Abrechnen von IT-Dienstleistungen über ein Netz. Angebot und Nutzung dieser Dienstleistungen erfolgen dabei ausschließlich über definierte technische Schnittstellen und Protokolle. Die Spannbreite der im Rahmen von Cloud Computing angebotenen Dienstleistungen umfasst das komplette Spektrum der Informationstechnik und beinhaltet unter anderem Infrastruktur (z. B. Rechenleistung, Speicherplatz), Plattformen und Software."* [15]

Cloud-Computing zeigt in Deutschland ein starkes Wachstum auf. Besonders in der Digitalisierung von Geschäftsprozessen setzen vorwiegend mittelständige Unternehmen auf die Cloud.[31]

Damit von einer Cloud gesprochen werden kann müssen bestimmte Merkmale im geschäfts- und technikbezogenen Bereich zutreffen. Auszeichnende geschäftsbezogene Merkmale sind die Bereitstellung und Nutzung von IT-Ressourcen als Service, schnelle und flexible Ressourcen-Verfügbarkeit, Ressourcen-Zuordnung durch den Nutzer, kurze Vertragsbindungen, minimale Vorabinvestitionen und Abrechnungen entsprechend der Nutzung der Ressourcen.
Technik bezogene Merkmale beziehen sich auf die flexible Bereitstellung skalierbarer IT-Ressourcen, eine gemeinsam nutzbare Infrastruktur, hohe Automatisierung, sowie Standardisierung, eine logisch zentralisierte und virtualisierte IT-Infrastruktur, Zugriff über den Browser, das Smartphone oder Tablets, vollständige lastabhängige Skalierbarkeit und die Messbarkeit des Verbrauchs der IT. Der Verbrauch beinhaltet Computerstunden, Speicherplatz und Rechenleistung . [27]

3.3.1 Cloud-Service-Ebenen

Da der Service einer Cloud in verschiedenen Formen angeboten wird, soll im Rahmen dieser Thesis abgegrenzt werden, welche Technologie und Form dafür geeignet ist. Dazu wird in diesem Kapitel die Cloud genauer beschrieben.

Nachfolgender Inhalt über die verschiedenen Cloud-Service-Ebenen referiert das Buch "Cloud Computing als neue Herausforderung für Management und IT", siehe hierzu [27].

Cloud-Computing wird in Service-Ebenen unterteilt. Das National Institute of Standards and Technology (NIST) beschreibt ein 3-Ebenen-Modell bei dem für den Cloud-Anwender unterschiedliche, nutzbare Cloud-Dienstleistungen zur Verfügung stehen. Diese 3 Service-Ebenen werden unterschieden in Software as a Service (SaaS) , Plattform as a Service (PaaS) und Internet as a Service (IaaS). Je nach bedarf kann der Anwender die für seine Geschäftsprozesse passende Cloud-Ebene als Service in Anspruch nehmen, wobei die Cloud-Ebenen auch aufeinander aufbauen können.

Im Einzelnen unterscheiden sich die Ebenen in folgenden Punkten:

- **Infrastructure as a Service (IaaS)**
  Bei diesem Service erhält der Cloud-Anwender Zugriff auf skalierbare Rechen-, Speicher- und Netzressourcen, wie in Abbildung 3.5. Diese werden als IT-Dienstleistung zur Verfügung gestellt und stehen unter der Verantwortung des Nutzers. Je nach Gebrauch können virtuelle Systeme mit unterschiedlichen Betriebssystemen zugeordnet und wieder freigeben werden. Im Vergleich zu der Infrastruktur in einem Unternehmen muss der Anwender keine Sorge tragen, ob physikalisch ausreichend Hardware oder Netzwerkkapazitäten zur Verfügung stehen. Hierbei wird der Central Processing Unit - Prozessor (CPU)-Memory-Verbrauch (Rechenleistung), GigaByte (GB)/Zeiteinheit (Speicher) oder das Datentransfervolumen (Netz) gemessen.

- **Platform as a Service (PaaS)**
  Auf dieser Ebene wird für Anwendungsentwickler und Software-Architekten ein Entwicklungsplattform als Service angeboten, siehe Abbildung 3.5. Es steht eine Umgebung mit Datenbank-Services, Services für Integration, Zugriffskontrolle, Sicherheit, Synchronisierung und Datenhaltung zur Verfügung. Das erleichtert die Software-Entwicklung, da die benötigte Umgebung nicht selbst implementiert werden muss. Die PaaS-Ebene ist damit eine effektive Basis für die SaaS-Services.

- **Software as a Service (Saas)**
  Diese Ebene der Cloud umfasst bereits fertige Anwendungen, die dem Anwender zur Verfügung gestellt werden. Es können benötigte Applikationsbausteine ausgewählt und für den Nutzen der Geschäftsprozesse als funktionierende Lösung kombiniert werden. Die Anwendungen übergehen als Nutzungsrecht nicht an den Anwender, sondern stehen auch anderen Nutzern zur Verfügung, mit denen die Software geteilt wird. Benötigte

Services der PaaS-Ebene und Infrastruktur-Ressourcen der IaaS sind dabei in der Regel mit inbegriffen.

In der Abbildung 3.5 sind die Zuständigkeiten der nutzbaren Bereiche für den Anwender der verschiedenen Ebenen verdeutlicht.

Abbildung 3.5: Separierung der Zuständigkeiten [34], bearbeitet

3.3.2 Cloud-Formen

Nachfolgender Inhalt über die verschiedenen Cloud-Formen referiert das Buch “Cloud Computing als neue Herausforderung für Management und IT”. [27]

Cloud-Systeme können unterschiedlich typisiert auftreten. Deshalb kann nicht von der “einen” Cloud gesprochen werden. Nach dem NIST gibt es 3 typische Ausprägungen für Cloud-Formen:

- **Public Cloud**
  Die "Public Cloud" wird von einem Dienstleister betrieben der im Besitz entsprechender Infrastruktur, Software und Speicherressourcen ist. Dieser bestimmt den Betrieb und die in Verbindung stehenden Sicherheitsaspekte. Darauf hat der Nutzer keinerlei Einfluss. Ebenso wenig auf den physischen Ort der Datenhaltung, der allerdings je nach Anbieter geografisch nach Zonen grob geregelt ist. Der Anwender hat in der Regel über das Internet Zugriff auf die Cloud. Unternehmen, die eine Cloud bei einem Dienstleister in Anspruch nehmen, teilen sich die Infrastruktur meist mit weiteren Nutzern.

- **Private Cloud**
  Die "Private Cloud" ist das Gegenteil zur "Public Cloud". Bei dieser Cloud-Form, betreibt ein Unternehmen auf Basis einer Cloud-Architektur die Cloud-Umgebung selbst. Dabei ist der Zugriff beschränkt und wird meist über ein Virtual Private Network (VPN) realisiert (weitere Informationen siehe [28]). Bei einer Private-Cloud hat das Unternehmen die vollständige Kontrolle der IT-Betriebsumgebung und kann diese auf das Unternehmensprofil zugeschnitten anpassen.

- **Hybrid Cloud**
  Eine Verknüpfung zwischen unterschiedlichen Cloudtypen wird als "Hybrid Cloud" bezeichnet. In Unternehmen überwiegt diese "Mischform" zunehmend, da zum einen für den internen Bedarf die Sicherheit in Form einer Private-Cloud notwendig ist, zum anderen gleichzeitig deren Kunden flexiblen Zugriff auf den Service in Form einer "Public Cloud" bereitgestellt werden soll.

3.4 ASP.NET MVC

ASP.NET Model-View-Controller (MVC) ist Teil des .NET Framework. Das .NET Framework ist eine Klassenbibliothek, die alle Funktionen zur Entwicklung für Desktop- und Internet-Anwendungen bietet. Ein Entwickler hat die Möglichkeit, die über 1000 Klassen des .NET Framework's zu verwenden. Microsoft bietet zu jeder Klasse eine ausführliche Beschreibung der Methoden und Funktionen an.
Die Entwicklungsumgebung dazu ist Visual Studio von Microsoft, mit den Sprachen C# und Visual Basic und der Common Language Runtime (CLR) als Laufzeitumgebung. [38]

ASP.NET MVC ermöglicht die Erstellung von serverseitigen Webanwendungen durch Aufteilung in 3 Komponenten: dem "Model", der "View" und dem "Controller " (MVC). Das Model beinhaltet Daten, einerlei von welchem Datentyp, oder in welcher Form. Die View erzeugt eine HTML-Webseite mit den bereitgestellten Daten des Models. Der Controller ist für die Logik und Erzeugung der Daten zuständig. [33]

Die Funktionalität und der Ablauf einer Anfrage sind im einzelnen wie folgt beschrieben und in Abbildung 3.6 dargestellt:

Eine Anfrage (Request) einer Webseite geht bei dem Controller ein. Dieser ist nichts weiter als eine Methode welche die Logik des Gerüstes beinhaltet. Der Controller kann weitere Klassen und Methoden einbinden und aufrufen, dazu gehören zum Beispiel auch Datenbanken-Abfragen. Der Controller ist für das Erzeugen von Daten zuständig welche durch die Anfrage angefordert wurden.
Was von einer Anfrage zurück kommt sind fast immer Daten. Dies können Daten in Form einer Liste aus einer Datenbank oder errechnete Daten sein. Diese Daten werden in das Model gepackt und der View bereitgestellt. Das Model besteht also lediglich aus den Daten, mittels derer mit dem Benutzer interagiert wird und übernimmt keine logischen Aufgaben.
Die Aufgabe der View ist es die bereitgestellten Daten des Models, in Form einer HTML-Webseite, zu erstellen. Diese wird an den Browser des Benutzers zurückgeliefert, durch den sie ihm dann angezeigt wird.
Durch MVC ist es möglich eine organisierte Struktur zu bewahren, die zur Übersicht von Webanwendungen und Webseiten beiträgt.[33]

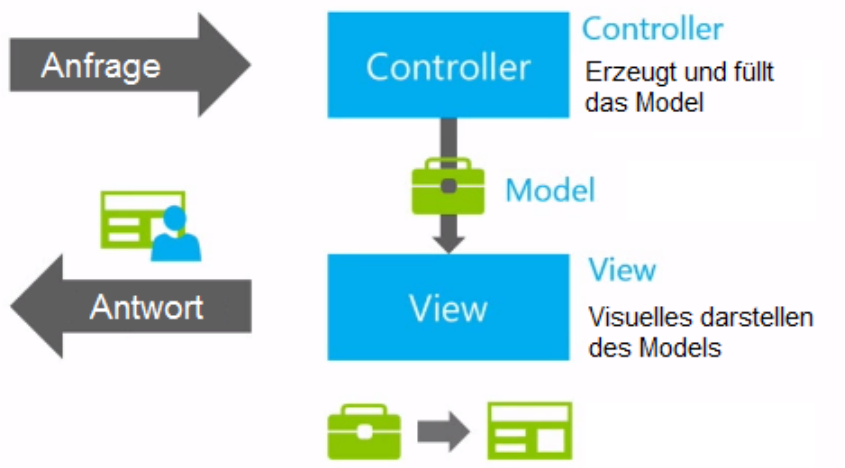


Abbildung 3.6: ASP.NET MVC Ablauf [33], bearbeitet

3.5 AES-Verschlüsselung

Der Advanced Encryption Standard(AES) gehört zu den Verschlüsselungsstandards bei dem sich noch keine Schwächen durch einen erfolgreichen Angriff auf den Algorithmus zeigten. Im Zuge eines vom NIST ausgeschriebenen Wettbewerbes, wurde 2001 der Rijndael Algorithmus zum Gewinner erklärt. Die Bezeichnung Rijndael ist eine Namensverknüpfung der

beiden Kryptologen und Entwickler von AES, Vincent Rijmen und Joan Daemen. In den Vereinigten Staaten von Amerika findet dieser Verschlüsselungsalgorithmus Zulassung für die höchsten Geheimhaltungsstufen. [35]

3.5.1 Verschlüsselungsprozess

Der Verschlüsselungsalgorithmus Rijndael kann mit einer Schlüssellänge von 128-, 192-, oder 256-Bit verwendet werden und besteht aus byteweisem Ersetzen, Vertauschen und XOR[3] Verknüpfungen. Die Blocklänge beträgt standardisiert 128-Bit und ist in Form einer 4 x 4 Matrix aus Byte-Variablen gegeben. Die 16 Bytes eines Klartextblocks werden dabei spaltenweise in die Matrix geschrieben.
Je nach Schlüssellänge wird eine unterschiedliche Anzahl an Runden zur Verschlüsselung durchgeführt, wie in Abbildung 3.7 dargestellt. Für eine Schlüssellänge von 128 Bit sind das 10, bei 192 Bit 12 und bei 256 Bit 14 Runden. Eine Runde beinhaltet eine Transformation durch Substitution und Permutation im Wechsel durch die Funktionen "SubBytes", Shiftrows", "MixColumns" und "AddRoundKey". Diese werden im einzelnen genauer erläutert. [35]

Der Ablauf erfolgt folgendermaßen, der 16 Byte-Klartextblock, als Block in Abbildung 3.7 gekennzeichnet, wird zunächst in der Initialisierungsrunde mit dem Schlüssel XOR Verknüpft. Danach wird je nach Schlüssellänge die Hauptrunde durchlaufen und mit der Hauptrunde die Funktionen "SubBytes", "Shiftrows", "MixColumns" und "AddRoundKey". Bei einem 128 Bit Schlüssel sind das 9 Runden. Für jede Runde wird zusätzlich der Schlüssel für die "AddRoundKey-Funktion" expandiert, siehe Kapitel 3.5.6, Schlüssel-Expansion. In der letzten, der zehnten Runde, wird "MixColumns" nicht mehr durchgeführt. Dies ist bei allen Schlüssellängen gleich, ob 128- 192- oder 256-Bit Verschlüsselung. Das Endprodukt ist dann der verschlüsselte Block.
Für der Entschlüsselung wird der Algorithmus rückwärts abgelaufen, dabei werden die Teilschlüssel umgekehrt zugeordnet. [36]

[3]XOR (exklusives Oder)ist eine logische Verknüpfung.

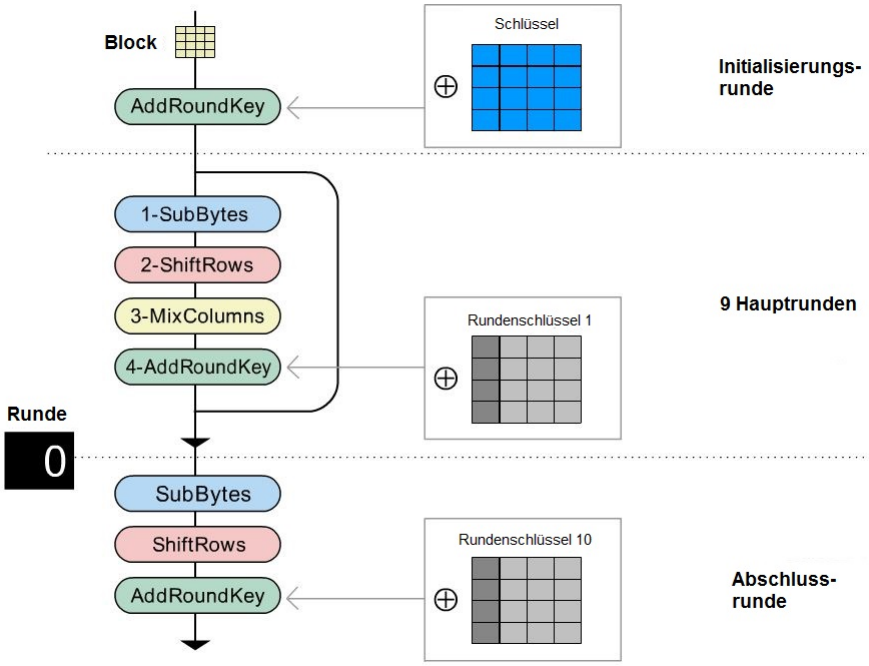


Abbildung 3.7: AES - Verschlüsselungs Ablauf [24], bearbeitet

3.5.2 AddRoundKey

Bei der Funktion AddRoundKey wird jeweils der 16 Byte-Block-State mit einem 16 Byte-Block des Schlüssels durch XOR verknüpft. Der 16 Byte-Block des Schlüssels ist dabei in jeder Runde ein andere, siehe Kapitel 3.5.6, Schlüssel-Expansion. [24]

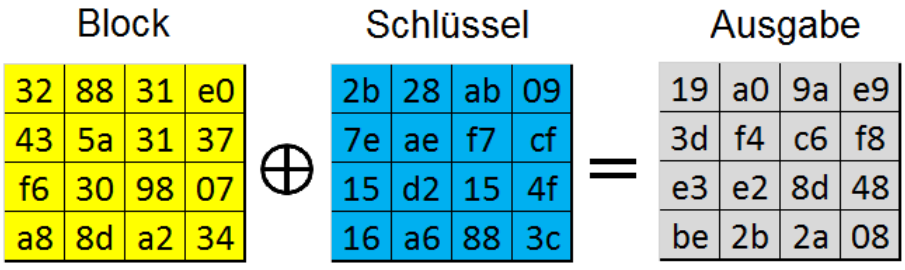


Abbildung 3.8: AES - AddRoundKey XOR Verknüpfung [24], bearbeitet

Das erste Byte 32_h in Hexadezimal[4] besteht aus 8-Bit welches sich binär[5] folgendermaßen zusammensetzt:

$$32_h = \underbrace{0011_b}_{3_h} \underbrace{0010_b}_{2_h} \text{ (1 Byte)}$$

[24]

Jedes Byte aus dem Block wird mit dem jeweiligen Byte des Schlüssels XOR verknüpft:

$$32_h \oplus 2b_h = 19_h$$
$$43_h \oplus 7e_h = 3d_h$$
$$\text{... usw.}$$

[24]

3.5.3 SubBytes

SubBytes, siehe Abbildung 3.7, wird nach der Initialen Runde 9 mal in der Hauptrunde durchgeführt und noch einmal in der zehnten, abschließenden Runde, bei einem 128 Bit Schlüssel. Bei einer Schlüssellänge von 192 Bit wären es 11 Hauptrunden und eine finale Runde. Bei der Schlüssellänge von 256 Bit, 13 Hauptrunden und eine finale Runde.

Die Operation SubBytes realisiert eine Konfusion durch eine nichtlineare Substitution. Mithilfe einer Substitutions-Box werden die 16 Bytes in der 4 x 4 Matrix ersetzt, wie in Abbildung 3.9 dargestellt. Die Substitutions-Box ist demnach eine Ersetzungstabelle. Dabei wird wie folgt vorgegangen. Der Hexadezimalwert für $S_{1,1} = \{53_h\}$ wird anhand der Ersetzungstabelle der neue Wert durch den Reihen-Index '5' und den Spalten-Index '3' definiert. Nach Abbildung 3.10 ist $S'_{1,1} = \{ed_h\}$. [40]

[4]Hexadezimalsystem ist ein Zahlensystem mit der Basis 16. Es gelten die Werte 0 bis 9 und A bis F.
[5]Binäreszahlensystem ist ein Zweiersystem, bei dem es nur die Werte 0 oder 1 gibt.

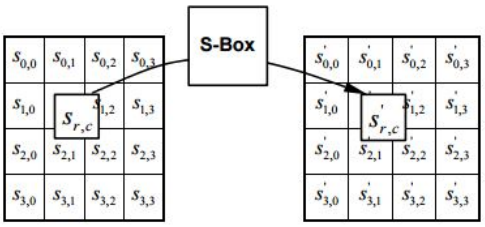


Abbildung 3.9: AES - SubBytes Zuweisung der Substitutions-Box [40]

x \ y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Abbildung 3.10: AES - SubBytes Subtitutions-Box [40]

Die Substitutions-Box ist nicht anfällig auf differenzielle und lineare Kryptoanalysen. Zu der Berechnung der Werte der Substitutionsbox gibt es weitere Informationen in der Federal Information Processing Standards Publication (FIPS PUBS) 197 vom 26. November 2001 über AES vom NIST. [40]

3.5.4 ShiftRows

Nachfolgender Inhalt über AES, zu dem Kapitel 3.5.4, wurde der Federal Information Processing Standards Publications (FIPS PUBS) 197, vom 26. November 2001 entnommen. [40]

Nachdem ”Konfusion” durch SubBytes in Kapitel 3.5.3 betrieben wurde, folgt die lineare Funktion ”ShiftRows”, welche die Zeilen der 4 x 4 Matrix mischt. Ebenso wie ”Subbytes” wird ”ShiftRows” in jeder Hauptrunde vertreten und einmal in der finalen Runde. Dabei ändern sich je nach Schlüssellänge die Durchgänge in der Hauptrunde.
ShiftRows erzielt Diffusion durch Permutation. Die erste Zeile der Matrix $r = 0$ wird nicht rotiert. Die zweite Zeile wird um 1, die dritte Zeile um 2 und die 4 um 3 nach links rotiert, wie in Abbildung 3.11 dargestellt.

$S'_{r,c} = S_{r,(c+shift(r,NB))modNB}$ für $0 < r < 4$ und $0 \leq c < NB$

$shift(r, NB)$ besteht aus der Zeilennummer r, und der Spaltenanzahl $NB = 4$.

$shift(1, 4) = 1$; $shift(2, 4) = 2$; $shift(3, 4) = 3$

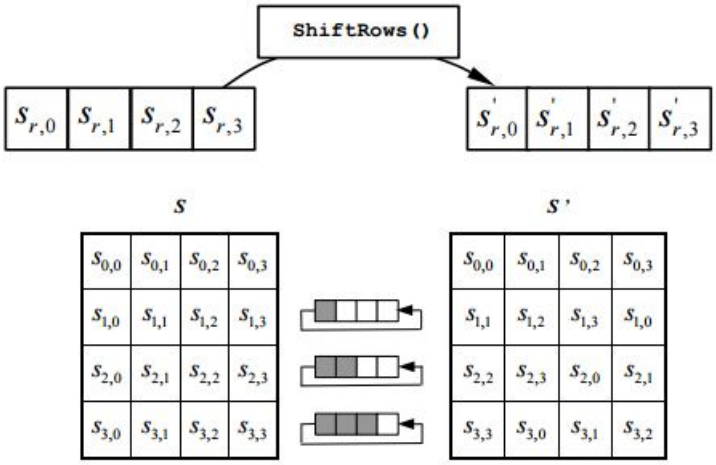


Abbildung 3.11: AES - ShiftRows - mischen der Bytes durch Rotation [40]

3.5.5 MixColumns

MixColumns trägt ebenfalls zur Diffusion bei. Allerdings wird nicht wie bei ShiftRows 3.5.4 Fokus auf die Zeilen gelegt, sondern die Spalten der Matrix gemischt. Dazu wird eine Multiplikation mit folgender Matrix durchgeführt:

$$\begin{pmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \cdot \begin{pmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{pmatrix} \quad für \quad 0 \leq c < NB \tag{3.1}$$

$$Beispiel für: S'_{0,c} = (\{02\} \cdot S_{0,c}) \oplus (\{03\} \cdot S_{1,c}) \oplus S_{2,c} \oplus S_{3,c} \qquad [40] \tag{3.2}$$

MixColumns ist die einzige Funktion, welche lediglich in den Hauptrunden stattfindet und nicht mehr in der finalen Runde verwendet wird. Das liegt daran, dass bei einer letzten Runde nicht noch mehr Sicherheit durch Vertauschen der Reihenfolge der Schlüsseladdition und

MixColumns entstehen kann. [35]

3.5.6 Schlüssel-Expansion

Die Schlüsselexpansion generiert für jede Runde einen neuen 16-Byte-Block der für "AddRoundKey", wie in Kapitel 3.5.2 beschrieben, verwendet wird. Hier ist die Schlüsselgröße von Bedeutung. Bei einer Schlüssellänge von 128 Bit, wird der Schlüssel auf 176 Byte expandiert. Bei 192 Bit, auf 208 Byte und bei 256 Bit auf 240 Byte.

Bei einem 128-Bit-Schlüssel und einer 128-Bit Blockgröße, werden insgesamt 11 Schlüssel mit je 16 Byte benötigt. Der Original Schlüssel in der Initialisierungsrunde, weitere 9 Schlüssel für die Hauptrunde und einen zehnten in der finalen Runde. Das sind zusammen 11 Schlüssel mit je 16 Byte und ergibt einen Gesamtschlüssel von 176 Byte. Dasselbe gilt für 192- und 256-Bit-Schlüssel mit der jeweiligen Rundenanzahl. [35]

Die Expansion der nächsten 4 Byte des nächsten Schlüssels wird folgendermaßen durchgeführt: Hierbei kommen 3 Funktionen zum Gebrauch: "RotWord", "SubWord" und "Rcon". RotWord führt eine byteweise Linksrotation der vier Bytes durch, wie in der Rechnung 3.3. SubWord ersetzt jedes der vier Bytes mithilfe der Subtitutions-Box, welche auch im Kapitel 3.5.3 verwendet wird. Rcon ist konstant und besteht aus 4 Bytes, wobei die letzten 3 Bytes stets null sind. Der Wert des 1. Bytes enthält eine Zweierpotenz die anhand der Rundenanzahl berechnet wird. In der nachfolgenden Abbildung 3.12 wird die Rcon-Tabelle dargestellt: [35]

Rcon

01	02	04	08	10	20	40	80	1b	36
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00

Abbildung 3.12: AES - Schlüssel-Expansion - Rcon-Tabelle [24]

Die letzten 4 Bytes werden zunächst durch RotWord rotiert:

$$\begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \Rightarrow \begin{bmatrix} a_{0,3} \\ a_{1,3} \\ a_{2,3} \\ a_{3,3} \end{bmatrix} RotWord \rightarrow \begin{bmatrix} a_{1,3} \\ a_{2,3} \\ a_{3,3} \\ a_{0,3} \end{bmatrix} \tag{3.3}$$

[24]

Anschließend werden die 4 Bytes mithilfe der Subtitutions-Box, ersetzt:

$$\begin{bmatrix} a_{1,3} \\ a_{2,3} \\ a_{3,3} \\ a_{0,3} \end{bmatrix} \rightarrow \begin{bmatrix} S - Box \end{bmatrix} \rightarrow \begin{bmatrix} a'_{1,3} \\ a'_{2,3} \\ a'_{3,3} \\ a'_{0,3} \end{bmatrix} \tag{3.4}$$

[24]

Die 4 Bytes werden dann mit dem Rcon Block und den 4 folgenden Bytes, 16 Bytes des Schlüssels zuvor, XOR-Verknüpft. Das Ergebnis sind die ersten 4 Bytes des neuen Runden-Schlüssels:

$$\begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \Rightarrow \begin{bmatrix} a_{0,0} \\ a_{1,0} \\ a_{2,0} \\ a_{3,0} \end{bmatrix} \oplus \begin{bmatrix} a'_{1,3} \\ a'_{2,3} \\ a'_{3,3} \\ a'_{0,3} \end{bmatrix} \oplus \begin{bmatrix} 01 \\ 00 \\ 00 \\ 00 \end{bmatrix} = \begin{bmatrix} b_{0,0} & 00 & 00 & 00 \\ b_{1,0} & 00 & 00 & 00 \\ b_{2,0} & 00 & 00 & 00 \\ b_{3,0} & 00 & 00 & 00 \end{bmatrix} \tag{3.5}$$

[24]

Die Folgenden 3-Byte-Blöcke werden dann errechnet, indem die 4 zuletzt errechneten Bytes, mit den 4 folgenden Bytes, 16 Bytes zuvor, XOR-Verknüpft werden.

$$\begin{bmatrix} a_{0,1} \\ a_{1,1} \\ a_{2,1} \\ a_{3,1} \end{bmatrix} \oplus \begin{bmatrix} b_{0,0} \\ b_{1,0} \\ b_{2,0} \\ b_{3,0} \end{bmatrix} = \begin{bmatrix} b_{0,0} & b_{0,1} & 00 & 00 \\ b_{1,0} & b_{1,1} & 00 & 00 \\ b_{2,0} & b_{2,1} & 00 & 00 \\ b_{3,0} & b_{3,1} & 00 & 00 \end{bmatrix} \tag{3.6}$$

[24]

Der erzeugte Runden-Schlüssel wird in der Funktion "AddRoundKey", wie in Kapitel 3.5.2 erklärt, verwendet. Für den Nächsten Runden-Schüssel wird die beschriebene Schlüssel-Expansion erneut durchlaufen.

Diese Vorgehensweise ermöglicht eine sichere Verschlüsselung, die im Rahmen dieser Thesis verwendet werden soll, um die Patientendaten auf diese Weise Datenschutzkonform auszulagern.

4 Konzept

In diesem Kapitel werden zunächst das Ziel und die Anforderungen beschrieben. Es folgen die Anwendungsansätze sowie die Evaluation der Cloud-Systeme und deren Betrachtung. Somit kann eine Entwurfsentscheidung bestimmt werden auf die eine Beschreibung des Systems und den Komponenten im einzelnen folgt.

4.1 Ziel

Das Ergebnis diese Arbeit ist die Implementierung eines Prototypen, um Patienten ihre Röntgenbilder auf einer Onlineplattform bereitstellen zu können.

4.2 Datenschutz Einschränkungen

Der Datenschutz innerhalb eines Unternehmens muss rechtskonformes Verhalten gewährleisten. Gehen Daten verloren oder werden sie an Dritte weitergegeben, sind Strafen und Rufschäden die Folge. Als Sensibel gelten Daten von Kunden, Lieferanten, Mitarbeitern, oder interne Daten wie Finanzen oder Erfindungen. Nach dem Bundesdatenschutzgesetz (BDSG) sind vor allem personenbezogene Daten zu schützen, zu denen Patientendaten zählen. Da in dieser Thesis Bilddaten einer Zahnarztpraxis mithilfe eines Drittanbieters bereitgestellt werden, müssen gewisse Richtlinien eingehalten werden. In diesem Kapitel werden die allgemeinen gesetzlichen Regelungen für Patientendaten und die Möglichkeiten der Auslagerung definiert. [42] [47]

4.2.1 Patientendaten

Patientendaten entstehen bei der Behandlung eines Patienten durch einen Arzt, oder entsprechendes Personal. Dabei wird ein Behandlungsvertrag geschlossen, welcher durch das Bürgerliches Gesetzbuch (BGB) § 630a ff. geregelt ist. Unter anderem werden dem Patienten die Rechte auf eine unverzügliche Einsichtnahme der Patientenakte und Daten nach Verlangen eingeräumt. Außerdem müssen die erhobenen Patientendaten, wie die Dokumentation der Behandlung, für die Dauer von 10 Jahren nach Abschluss der Behandlung aufbewahrt werden. Für Röntgenbilder gelten dabei allerdings abweichende Regelungen wie in der Röntgenverordnung (RöV) § 28 Abs. 3. [4] [2]

Patientendaten sind nach den Begriffsbestimmungen § 3 Abs. 1, 9 BDSG besondere Daten und unterliegen somit einem erhöhten Schutz. Der Arzt muss im Sinne des Datenschutzes eine sichere Erhebung, Speicherung, Veränderung und Übermittlung der Daten gewährleisten. Verstöße stellen Verletzungen des Datengeheimnisses nach § 5 BDSG dar und führt nach § 4 BDSG zu einer Untersagung der Datenverarbeitung. Das Strafgesetzbuch (STGB) regelt außerdem die Verletzungen von Privatgeheimnissen und der ärztlichen Schweigepflicht in § 203 Abs. 1 wie folgt: [47] [3] [16]

> *"(1) Wer unbefugt ein fremdes Geheimnis, namentlich ein zum persönlichen Lebensbereich gehörendes Geheimnis oder ein Betriebs- oder Geschäftsgeheimnis, offenbart, das ihm als Arzt, Zahnarzt, Tierarzt, Apotheker oder Angehörigen eines anderen Heilberufs, der für die Berufsausübung oder die Führung der Berufsbezeichnung eine staatlich geregelte Ausbildung erfordert, anvertraut worden oder sonst bekanntgeworden ist, wird mit Freiheitsstrafe bis zu einem Jahr oder mit Geldstrafe bestraft."* [5]

Die ärztliche Schweigepflicht dient als Grundlage des Vertrauensverhältnisses zwischen einem Patienten und dem Behandelnden. Sie gilt auch gegenüber anderen Ärzten, Familienangehörigen des Betroffenen und des Arztes und bleibt nach dem Tod des Patienten weiterhin bestehen. Der Patient kann durch eine ausdrückliche Einwilligung aus freiem Willen, den Arzt zu bestimmten Zwecken von dessen Schweigepflicht befreien. Vergleich dazu § 4a BDSG. [3] [1] [16]

4.2.2 Auslagern von Patientendaten

Das Auslagern von Patientendaten bei einem Drittanbieter ist im allgemeinen zulässig, solange § 11 Abs. 1 im BDSG zur Erhebung, Verarbeitung oder Nutzung personenbezogener Daten im Auftrag erfüllt ist:

> *"(1) Werden personenbezogene Daten im Auftrag durch andere Stellen erhoben, verarbeitet oder genutzt, ist der Auftraggeber für die Einhaltung der Vorschriften dieses Gesetzes und anderer Vorschriften über den Datenschutz verantwortlich. Die in den §§ 6, 7 und 8 genannten Rechte sind ihm gegenüber geltend zu machen."* [3]

Paragraphen 6, 7 und 8 betreffen die Rechte des Betroffenen, und den Schadensersatzanspruch bei Datenmissbrauch. Die Patientendaten betreffenden gesetzlichen Regelungen, die in Kapitel 4.2.1, Patientendaten, zuvor definiert wurden, müssen umgesetzt werden. Bei der Auslagerung der Daten ins Ausland, wobei andere Mitgliedstaaten außerhalb der Europäische Union (EU) gemeint sind, kann nach den Vorgaben der EU und des deutschen Gesetzgebers nicht von einem gleichwertigem Datenschutzniveau ausgegangen werden. Somit liegt die Übermittlung und Auslagerung der Daten in der Verantwortung des Übermittlers nach

Paragraph 4b des BDSG. [16] [3]

Für das Auslagern von Patientendaten sollte eine Pseudonymisierung der Bilddaten vorangestellt werden. Genauer definiert wird dies unter § 3 Abs. 6a BDSG:

> *"(6a) Pseudonymisieren ist das Ersetzen des Namens und anderer Identifikationsmerkmale durch ein Kennzeichen zu dem Zweck, die Bestimmung des Betroffenen auszuschließen oder wesentlich zu erschweren."* [3]

4.2.3 ISO 27001 Zertifizierung

Die ISO 27001 Zertifizierung ist eine internationale Norm für Informationssicherheits - Managementsysteme (ISMS). Die Bedrohung durch Hackerangriffe, Datenverlust oder Missbrauch vertraulicher Informationen kann immense Imageschäden und Kosten verursachen. Durch die Zertifizierung der ISO 27001 ist ein Ansatz für den Schutz und Integrität von vertraulichen Daten gegeben. Die Durchführung wird von einer unabhängigen und anerkannten Instanz getätigt, wie zum Beispiel von zertifizierten Auditoren des BSI . Ein Auditor muss die Referenzdokumente der zu zertifizierenden Institution sichten, sowie eine Prüfung vor Ort durchführen und ein Audit-Report erstellen. Dieser Audit-Report wird der Zertifizierungsstelle vorgelegt welche diesen prüft.
Eine Zertifizierung nach der ISO 27001 erfüllt einen vertrauenswürdigen Nachweis der Realisierung ergriffener Maßnahmen nach der IT-Grundschutz-Vorgehensweise. Nicht zuletzt durch den "Plan-Do-Check-Act-Zyklus" (PDCA). Dieser umfasst vier Schritte zur ständigen Verbesserung der Prozesse und Abläufe in einem Unternehmen und werden nachfolgend erklärt, sowie in Abbildung 4.1 dargestellt.

- **Plan** umfasst die Identifizierung und Planung relevanter Prozesse, die Abgrenzung der Anforderungen und das Ableiten der Ziele und Maßnahmen.
- **Do** beschreibt die Umsetzung der geplanten Maßnahmen und Organisatorischen Aspekte.
- **Check** definiert die Prüfung, Überwachung und Bewertung der Prozesse und Regelungen.
- **Act** umfasst das Reflektieren der Prozesse um Folgemaßnahmen und Verbesserungen anzustoßen oder Korrekturen einzuleiten.

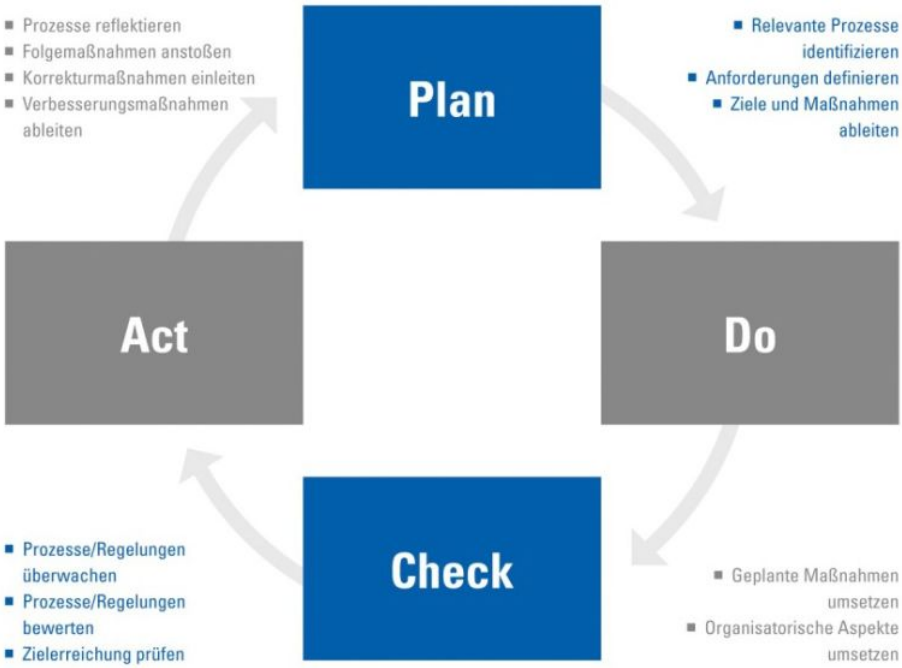


Abbildung 4.1: Datenschutz - PDCA Zyklus [50]

Die Vorteile für ein Unternehmen durch eine Zertifizierung nach der ISO 27001 sind:

- kontinuierliche Kontrolle und Optimierung der Informationssicherheit durch ständiges Anpassen und Verbessern nach dem PDCA-Zyklus,
- Risikominimierung durch ein strukturiertes ISMS welches zum Erkennen von Bedrohungen beiträgt,
- Datensicherheit durch Schutz vor Missbrauch oder Verlust,
- Informationssicherheit im Unternehmen, da die ISO 27001 Zertifizierung über alle Hierarchieebenen gestellt wird und durch das Absolvieren verschiedener Trainings und Schulungen,
- Erfüllen externer Anforderungen durch Berücksichtigung der Verfügbarkeit, Integrität und Vertraulichkeit,
- Vertrauen gegenüber anderen Unternehmen, Kunden und Geschäftspartnern.

[50] [37]

4.3 Anforderungen

Anforderungen für den Entwurf sind im Folgenden aufgelistet:

- **Geringe Hemmschwelle bei Benutzung:** Für den Patienten sollen keine Hemmnisse zum Erlangen der Bilddaten entstehen. Dies beinhaltet keine Registrierung oder Passworteingabe beim Herunterladen. Außerdem sollen keine weiteren Kosten für den Patienten entstehen.
- **Internetzugang:** Der Patient benötigt einen Internetzugang über Mobile Daten oder WLAN.
- **Personenbezogene Daten:** Es soll kein Zugriff auf personenbezogene Daten erfolgen oder Code implementiert werden, welcher dem Patienten durch abfotografieren des QR-Codes schaden könnte.
- **Alternative:** Alle Patienten, die nicht der entsprechenden Zielgruppe zutreffen, sollen die Möglichkeit haben, einen Link zum Herunterladen der Bilddaten per E-Mail zu bekommen, sofern eine E-Mail Adresse vorhanden ist.
- **Verfügbarkeit:** Die Verfügbarkeit der Systeme sowie der Einsatz sollen langfristig gewährleistet sein.
- **Bildqualität:** Die Bilddaten sollen in einer minimalen Auflösung von 1024 x 768, mit einer Größe von 100 KiloByte (KB) bis 250 Kilobyte und in der Originalauflösung von ca. 1920 x 1080 mit einer maximalen Größe von 1 MegaByte (MB) angeboten werden.
- **Plattformunabhängig:** Die Bereitstellung der Bilddaten soll Plattformunabhängig im JPEG Format erfolgen. Der QR-Code soll ebenfalls von jedem standartmäßigem QR-Code Reader gelesen werden können.
- **Integrität:** Die Bilddaten sollen mit AES 256 bit verschlüsselt werden (siehe auch Kapitel Grundlagen 3.5). Jeder Patient bekommt für sein Bilddaten einen eigenen Schlüssel zum Entschlüsseln. Es soll lediglich der Person möglich sein, die Bilddaten einzusehen, welche den entsprechenden Link oder QR-Code besitzt. Das Verschlüsselungsverfahren sollte nicht auf dem Cloud-System stattfinden, sondern beim Übermittler der Bilddaten. Die Schlüssel dazu dürfen nicht auf der Cloud abgelegt werden.
- **Vertraulichkeit:** Die Verbindung beim Hochladen, sowie beim Herunterladen soll durch Hypertext Transfer Protocol Secure (HTTPS) gesichert sein.
- **Bereitstellung:** Wenn es der Patient erlaubt, sollen die Bilddaten für bis zu 7 Tage in der Cloud bereitgestellt werden, nachdem diese hochgeladen wurden. Nach Ablauf dieser Zeit werden diese wieder entfernt. Es soll im Nachhinein möglich sein, den Zeitraum über die Verfügbarkeit der Bilddaten auf der Cloud anzupassen.

- **Entfernen:** Die Bereinigung der Bilddaten soll einmal täglich automatisiert, Serverseitig ablaufen. Der Patient hat die Möglichkeit, die Bilddaten bereits jederzeit früher zu entfernen.
- **Datenschutz:** Eine ISO 27001 Zertifizierung des Cloud-Anbieters muss vorhanden sein. Die Rechte der hochgeladenen Bilddaten sollen nicht an den Anbieter der Cloud abgetreten werden. Der Patient kann die Datenschutzerklärung einsehen und wird vor dem Abfotografieren des QR-Codes durch eine Info in Form einer Checkbox darauf hingewiesen. Mit Aktivieren der Checkbox, erklärt sich der Patient mit den Datenschutzbestimmungen zur Auslagerung der Bilddaten einverstanden. Ohne Einverständnis des Patienten, werden die Bilddaten nicht auf die Cloud hochgeladen.
- **Lokalisierung:** Der Patient bekommt die Benutzeroberfläche der HTML-Seite in eine der folgenden Sprachen angezeigt:
  - Deutsch
  - Englisch

  Es ist nicht spezifiziert ob weitere Sprachen nachträglich ohne Code-Änderungen ergänzt werden können.
- **Performance:** Bei einem Internetzugang mit einer Geschwindigkeit zum Hochladen von 576 kbit/s und Bilddaten mit der Größe von insgesamt 750 kb, beträgt die Wartezeit maximal 10 Sekunden. Die Erstellung des Qr-Codes soll unter 10 Millisekunden betragen. Bei einem Internetzugang mit einer Geschwindigkeit zum Herunterladen von 6016 kbit/s, soll die Wartezeit der Bilddaten, mit der Größe von insgesamt 750 kb, 997 Millisekunden betragen. Das Löschen der Bilddaten hängt von der Belastung des Webservers ab. Im Durchschnitt benötigt ein Server mit 1 CPU und 1 GB Random-Access Memory - Arbeitsspeicher (RAM) für die Durchführung einer Anfrage $< 0{,}1$ Millisekunden.

4.4 Anwendungsansätze Cloud-Systeme

Um eine Evaluierung verschiedener Cloud-Systeme vorzunehmen, sind Anwendungsansätze notwendig. Diese werden in folgendem Kapitel definiert und dienen als Grundlage für Berechnungen und Entscheidungen im weiteren Verlauf.

4.4.1 Grundfaktoren

Um genauere Berechnung der Kosten durchzuführen, sind einige Grundinformationen notwendig. Um das Interesse und den Nutzen zu bestimmen, werden Schätzungen des Produktmanagers verwendet. Weitere Zahlen werden Aufgrund von Tests oder Tatsachen entnommen. Alle Berechnungen und Schätzungen beziehen sich auf ein Jahr, ab Beginn der Markteinführung.

Nutzen
Für genauere Informationen über das Interesse und den Nutzen zum plattformunabhängigen Austausch von Bilddaten, wurde die Produktmanagerin Fr. Dr. Amor der SIDEXIS 4 Software befragt. Nach ihrer Kenntnis werden geschätzt 20 Prozent der Patienten einer Praxis Interesse an diesem Dienst zeigen und der Nutzen bei einem Wert von 1 Prozent im ersten Jahr liegen, mit steigender Tendenz.

Patienten
Die Anzahl der Patienten, werden aus einer Datenbank einer realen Praxis abgeleitet. Diese Praxis verfügt über 4369 Patienten. Bei einer Patientennutzung von einem Prozent, ergibt das ein Wert von 43,6 Patienten pro Praxis.

Praxen
Die Anzahl der Praxen, welche über die Software SIDEXIS 4 verfügen, beläuft sich auf 100-150 Installationen. Da sich SIDEXIS 4 allerdings noch in der Markteinführung befindet, wurde für weitere Berechnungen mit einer Anzahl von 1000 Installationen gerechnet.

Bildgröße
Die Bilder, welche im Joint Photographics expert Group (JPEG) Format hochgeladen werden, haben zwei unterschiedliche Größen. Ein Bild wird mit einer Auflösung von 1024 x 768 Angeboten, das zweite in einer höheren Auflösung von 1920 x 1080. Bilder mit der niedrigeren Auflösung haben im Durchschnitt ungefähr 150 KB. Bilder mit der höheren Auflösung haben ca. 600 KB. Das entspricht einer Gesamtgröße beider Bilder von 0,75 MB.

Zusammengefasst ergeben das folgende Grundlagen für weitere Berechnungen:

Grundlagen	
Nutzen in %	1
Patienten pro Praxis	4369
Praxen	1000
Gesamt Bildgröße in MB	0,75

Tabelle 4.1: Grundfaktoren zur Kostenberechnung

4.4.2 Speicherplatzbedarf

Anhand den Durchschnittswerten, von den Auflösungen der Bilder und den Zahlen der Patienten, im Bezug auf den Nutzen der Patientenkommunikation, kann der benötigte Speicherplatz errechnet werden. Der Speicherplatzbedarf ist ein wichtiger Faktor. Es muss jederzeit genug Platz auf der Cloud vorhanden sein, um die Bilddaten dort speichern zu können. Der Bedarf an Speicherplatz beeinflusst wiederum die laufenden Kosten der Cloud.

Es ergeben sich folgende Berechnungen:

1% von 4369 Patienten	=	43,6 Patienten
43,6 Patienten * 0,75 MB	=	32,7 MB
1000 Praxen * 32,7 MB	=	32700 MB
32700 MB in 1 Monat	=	2725 MB

Somit beträgt der durchschnittliche Speicherplatzbedarf pro Monat 2,7 GB. Wenn man davon ausgeht, dass alle potentiellen Patienten am selben Tag in die Praxis kommen und die Ressource nutzen, so wäre der Worst Case[1] ein Speicherplatzbedarf von 32,7 GB. Diese würden dann für bis zu sieben weitere Tage zum Herunterladen zur Verfügung gestellt werden und Speicherplatz belegen. Damit ist jedoch nicht zu rechnen. Allerdings sollte der Speicherplatz der Cloud zumindest die Hälfte des Speicherplatzbedarfes des Worst Case betragen. Dieser liegt bei 16,35 GB.

4.4.3 Cloud-Systeme

Die Funktionen der Cloud spielen eine wichtige Rolle für die Auswahl eines Systemes. Zur weiteren Evaluierung stehen Microsoft Azure, Amazon und Strato zur Auswahl, da diese die entsprechenden Anforderungen erfüllen. Jeder dieser Anbieter bietet unterschiedliche Tarife, zu unterschiedlichen Preisen und Leistungen an. Für diese drei Anbieter werden die Kriterien

[1] Worst Case: ungünstigster-anzunehmender-Fall

der Technologie, dem Datenschutz, der Kosten, sowie der Zukunftssicherheit genauer evaluiert. Dadurch kann im Anschluss entschieden werden, welches der Systeme für den weiteren Verlauf am geeignetsten ist.

Weitere Anbieter wie Cloudinary[2] die in dieser Thesis nicht weiter behandelt werden, schieden im Vorfeld aus, da sie den Anforderungen des Projektes nicht gerecht werden konnten. Cloudinary bietet Speicherplatz für das Hochladen von Bildern an. Mit Hilfe einer API sind diverse Modifikationen möglich. Auch der Verbindungsaufbau erfolgt verschlüsselt über ein Verschlüsselungsprotokoll Transport Layer Security (TLS). Die Bilddaten liegen allerdings unverschlüsselt auf dem Server. Bei dem Versuch die Bilder verschlüsselt als binäre Dateien hochzuladen, werden diese verworfen. Zudem finden sich keinerlei Informationen einer Datenschutzerklärung.
Aufgrund dieser Tatsachen konnte dieser und weitere Anbieter nicht zur Evaluation der Cloud-Systeme verwendet werden.

[2]Anbieter Webseite: http://cloudinary.com

4.5 Evaluation der Cloud-Systeme

Für die Evaluation, werden die Anbieter der Cloud-System anhand der Anwendungsansätze aus Kapitel 4.4, im Bezug auf die Anforderungen aus Kapitel 4.3 und den Einschränkungen aus Kapitel 4.2, detaillierter betrachtet.

4.5.1 Azure

Inhalte des nachfolgenden Kapitels 4.5.1 Azure wurden der Produktwebseite von Microsoft entnommen [14].

Technologie

Microsoft Azure stellt einen Cloud-Services bereit, der es ermöglicht, hochverfügbare, skalierbare Anwendungen und APIs zu schaffen. Unterstützend für Java, Node.js, Hypertext Preprocessor (PHP), Python, ASP.Net und Ruby, ist es auch möglich mit, Visual Studio[3] und dem integrierbaren Microsoft Azure Software Development Kit (SDK) zu entwickeln. Über ein Verwaltungsportal können Dienste, Webseiten, virtuelle Maschinen und Services überwacht, verwaltet und bereitgestellt werden. Administratoren können Ressourcen, Benutzerkonten und deren Kontingente und Preise konfigurieren und verwalten. Über eine Dienstverwaltungs-API als REST-API ist der Dienstzugriff und andere Integrationsszenarien möglich. Azure bietet als PaaS, als auch als Iaas für die Entwickler die Möglichkeit, die Infastruktur und die Services auf die Bedürfnisse ihrer Anwendung optimal anzupassen. Azure bietet neben dem Cloud-Dienst weitere Dienste zur Webseitenerstellung an. Insbesondere virtuelle Computer, Batchdienste, SQL-Datenbanken, einen Media-Service, Recovery-Service, Mobile-Service und weitere. Dadurch können die Dienste für Anwendungen miteinander kombiniert werden.

Datenschutz

Der Datenschutz verspricht durch Verpflichtung der International Organization for Standardization (ISO)/International Electrotechnical Comission (IEC) 27018 und ISO 27001 Sicherheit und Kontrolle der Daten. Mit Wertlegung auf Transparenz und Bereitstellung von Verschlüsselungsmöglichkeiten durch AES-256, sowie gesicherten Verbindungen der Übertragungen von Daten, erfüllt Microsoft die Vorgaben der EU-Datenschutzrichtlinien. [25]

[3]Visual Studio: Software als Entwicklungsumgebung für verschiedene Hochsprachen

Zukunftssicherheit

Im Bereich der Zukunftssicherheit ist in den nächsten Jahren keine Verschlechterung zu erwarten, wenn man den Kurs[4] des Unternehmens der letzten Jahre betrachtet. Als wirtschaftsstarkes Unternehmen ist langfristig mit Microsoft zu rechnen.[49]

Kosten

Microsoft Azure stellt verschiedene Hardware zur Verfügung. Diese umfasst die Anzahl der CPU, RAM und den Speicherplatz.

MT0	1 CPU, 0,75 GB RAM, 19 GB Speicher	0,0149€/ Stunde
MT1	1 CPU, 1,75 GB RAM, 224 GB Speicher	0,0596€/ Stunde
MT2	2 CPU, 3,50 GB RAM, 489 GB Speicher	0,1192€/ Stunde

Tabelle 4.2: Azure - Preise des Cloud-Services [12]

Zusätzlich kommen die Kosten für den erzeugten Datenverkehr. Dabei sind alle Transfers zum Server kostenlos. Unter einem Transfer zum Server versteht man in diesem Fall das Hochladen der Daten. Auch das Herunterladen ist pro Monat für die ersten fünf GB gebührenfrei. Danach werden bis zu zehn GB kosten von 0,0648€ pro GB anfallen.

Hochladen	unbegrenzt	kostenlos
Herunterladen	ersten 5 GB	kostenlos
Herunterladen	5 GB bis 10 GB	0,0648€ pro GB

Tabelle 4.3: Azure - Preise der Datenübertragung [13]

Jährliche Gesamtkosten
Aus den drei Varianten, Microsoft Tarif 0 (MT0), Microsoft Tarif 1 (MT1) und Microsoft Tarif 2 (MT2), können die jährlichen Kosten errechnet werden. Diese setzen sich aus dem Tarif und dem durchschnittlichen Speicherplatzbedarf von S1 = 2,73 GB zusammen:

$$\begin{aligned} \text{MT0} + \text{S1} &= 128{,}74€ \\ \text{MT1} + \text{S1} &= 514{,}94€ \\ \text{MT2} + \text{S1} &= 1024{,}89€ \end{aligned}$$

[4]Kurs der Microsoft Corp.: http://www.finanzen.net/aktien/Microsoft-Aktie

4.5.2 Amazon

Inhalte des nachfolgenden Kapitels 4.5.2 Amazon wurden der Produktwebseite von Amazon entnommen. [8]

Technologie

Amazon bietet mit Amazon Web Services (AWS) für die Cloud verschiedene Möglichkeiten an. Bestandteile von AWS ist Elastic Compute Cloud (EC2) als virtuelle Server mit Linux oder Windows betrieben. Der Arbeitsspeicher sowie die Prozessoren können jederzeit angepasst werden. Auch die Menge der Server kann beliebig variiert werden. Als Speicher stellt Amazon den Simple Storage Service (S3) zur Verfügung. Damit können beliebige Datenmengen in jedweder Form gespeichert werden.
Für Entwickler werden Simple Workflow Service (SWS), Simple Queue Service (SQS), Simple Email Service (SES) und Simple Notification Service (SNS) als Dienste bereitgestellt. Mit dem Service Elastic Beanstalk als Plattform, können Webanwendungen mit verbreiteten Hochsprachen wie Java, .NET, PHP, Node.js, python, oder Ruby entwickelt werden. Es stehen Datenbanken basierend auf Microsoft SQL Server oder MySQL zur Verfügung, sowie ein Identity and Access Management (IAM), zur Verwaltung von Benutzerkonten und Ressourcen.
Um Tarife und Ressourcen je nach Auslastung automatisch anzupassen, können EC2-Server mit Amazon Cloud Watch überwacht werden. Um Datenverlust vorzubeugen ist Amazon Glacier als Speicherservice der Daten einsetzbar welcher die Daten verschlüsselt sichert. [32]

Datenschutz

Dem Datenschutz in der Amazon Cloud wird höchste Priorität zugeschrieben. Der Kunde bekommt die Kontrolle über seine Daten, dem Standort, sowie dem Format der Speicherung und die Zugriffsberechtigung. Amazon nimmt am Safe Harbor-Programm[5] teil welche von den USA, der Europäischen Union und der Schweiz konzipiert wurden teil. Für Kunden der Europäischen Union in Benutzung mit AWS hat Amazon ein EU Datenschutz Whitepaper veröffentlicht.[10]

Zukunftssicherheit

Amazon mit stetig wachsendem Kurs[6] stellt für die Zukunftssicherheit keine Gefahr da. Als weltweit größter Online-Händler und weiteren vielfältigen Produkten lag der Nettoumsatz bei fast 89 Milliarden US-Dollar. Mit Amazon ist somit auch in längerfristig auf dem Markt

[5] Bei Datenschutzbeschwerden können nach dem Safe Harbor-Programm einem unabhängigen das Streitschlichtungsverfahren zugewiesen werden. Mehr Informationen unter: www.bbb.org/us/safe-harbor-complaints

[6] Kurs von Amazon: http://www.finanzen.net/aktien/Amazon-Aktie

zu rechnen. [48]

Kosten

Amazon stellt unterschiedliche Tarife zur Verfügung. Der Unterschied zu Azure ist in diesem Fall der, dass der Speicherplatz nicht fest vergeben ist, sondern dynamisch an den benötigten Speicherplatz angepasst wird. Dadurch gibt es für den Benutzer der Cloud keine Speicherplatzbegrenzung. Als Kosten werden lediglich die tatsächlich genutzten Leistungen erhoben.

AT0	1 CPU, 1 GB RAM	0,015$/ Stunde
AT1	1 CPU, 2 GB RAM	0,030$/ Stunde
AT2	2 CPU, 4 GB RAM	0,060$/ Stunde

Tabelle 4.4: Amazon - EC2 Preise [9]

Amazon erhebt Gebühren für verwendeten Speicherplatz. Es gelten unterschiedliche Preisregelungen für bestimmte Anforderungen . Für “PUT”, “COPY”, “POST” und “LIST” Befehle werden 0,0054$ pro 1000 Anfragen berechnet. “GET” Anfragen und alle anderen Datentransfers von der Cloud in das Internet werden mit 0,0043$ pro 1000 Anfragen berechnet.

Speicher	1GB	0,119$/ Monat
Anforderung	PUT, COPY, POST, LIST	0,0054$ pro 1000 Anforderungen
Anforderung	GET und alle anderen	0,0043$ pro 1000 Anforderungen
Anforderung	DELETE	kostenlos
Übertragung	1 GB im Monat	kostenlos
Übertragung	1 - 10 TerraByte (TB)	0,09$/ Monat

Tabelle 4.5: Amazon - Preise der Datenübertragung [9]

Die Kosten der Anforderungen setzen sich wie folgt zusammen:

Bei einer Anzahl von 1000 Praxen, welche zu Beginn im Kapitel 4.4.1 festgelegt wurde, ergibt dies eine Anzahl von 43600 Patienten. Für jeden dieser Patienten wird die Bilddatei in zwei unterschiedlichen Formaten hochgeladen. Demnach entstehen 87200 PUT-Anforderungen. Lädt jeder Patient beide Bild-Formate herunter, so entsteht ebenfalls eine Anzahl von 87200 GET-Anforderungen. Verteilt auf ein Jahr entspricht das ca. 7266 GET- und PUT-Anforderungen pro Monat.

7,266 PUT-Anforderungen * 0,0054$	=	0,039$/ Monat
7,266 GET-Anforderungen * 0,0043$	=	0,031$/ Monat

Jährliche Gesamtkosten
Die jährlichen Gesamtkosten von Azure Tarif 0 (AT0), Azure Tarif 1 (AT1), und Azure Tarif 3 (AT2), mit den Anforderungen sowie Speicherplatzbedarf von S1 = 2,73 GB und Datenübertragung ergeben jährliche Kosten von:

AT0 + S1	=	136,21$	Kurs: 1,1095	122,77€
AT1 + S1	=	265,81$	Kurs: 1,1095	239,81€
AT2 + S1	=	525,01$	Kurs: 1,1095	473,20€

4.5.3 Strato

Inhalte des nachfolgenden Kapitels 4.5.3, Strato, wurden der Produktwebseite von Strato entnommen [20].

Die Strato AG wurde 1997 in Berlin gegründet und bot auf dem Markt verschiedene Angebote zum hosten von Webseiten und Webadressen an. Mit der Server Cloud kamen sie erstmals 2012 auf den Markt, nachdem sie bereits 2011 den Oline-Speicher als HIDrive erfolgreich präsentiert hatten. Nach heutigem Stand fungiert Strato als Tochtergesellschaft der Deutschen Telekom AG. [51]

Technologie

Strato bietet eine Server Cloud mit vollem Root-Zugriff[7] auf eine Virtuelle Maschine (VM)[8] und durch den Einsatz von Server und Speicher, soll die maximale Verfügbarkeit erreicht werden. Alle Systeme laufen redundant[9] und werden getrennt von der Server Cloud gesichert. Die Tarife sind flexibel anpassbar und es wird nur in Rechnung gestellt, was auch genutzt wird. Außer den Infrastruktur-Angeboten, ist es besonders für den Webseiten-Gebrauch möglich, Domains, Speicherplatz, Datenbanken, Anwendungen, Mail-Dienste, oder FTP-Zugänge einzurichten. [21]

[7] Root-Zugriff: Benutzerkonto mit vollständigen Zugriffsrechten.
[8] Ein Computer der nicht direkt auf Hardware Ressourcen ausgeführt wird, sondern von einem physisch vorhandenen Computer bereitgestellt wird.
[9] redundant: mehrfache Sicherung der Systeme sodass bei Datenverlust keine Informationen verloren gehen können.

Datenschutz

Strato ist zertifiziert durch den TÜV-Süd[10] mit der ISO 27001. Die Übertragung und Administrierung erfolgt verschlüsselt über das Secure Shell (SSH) Netzwerkprotokoll. Die Standorte der Server sind in Deutschland und unterliegen damit dem deutschen Datenschutzgesetz. Die Server Cloud sorgt zudem für Sicherheit durch Antivirensoftware und regelmäßigem Überprüfen auf Schwachstellen.

Zukunftssicherheit

In den Anfängen des Unternehmens, wurde Strato bereits an das Unternehmen TELES[11] verkauft. TELES verkaufte Strato weiter an die freenet AG[12], die letztendlich von der Deutschen Telekom übernommen wurde. Laut Berichten der Online-Computerzeitschrift Heise, hatte Strato Schwierigkeiten den Kunden die versprochenen Leistungen zu erfüllen und kämpften mit Serverausfällen. Als Tochtergesellschaft der Deutschen Telekom ist in Zukunft trotzdem mit weiterem Wachstum der Strato AG zu rechnen. [26][51][30]

Kosten

Strato hat ein Zahlungssystem in Form von Krediten. Je nach Einstellungen der Hardware Details, werden stündlich unterschiedlich Kredite berechnet. Kredite kann man in verschiedenen Tarifen oder Abonnements erwerben und sind nur für maximal einen Monat gültig. Kredite die am Ende des Monats noch vorhanden sind, werden nicht auf den Folgemonat gutgeschrieben, sondern verfallen.

Reichen die Kredite eines Prepaid-Paketes für einen Monat nicht aus, so werden alle weiteren Kredite mit dem Starter Paket von einem Cent abgerechnet.

1 CPU	1 Kredit pro Stunde
1 GB RAM	1 Kredit pro Stunde
100 GB Speicher	1 Kredit pro Stunde

Tabelle 4.6: Strato - Preise der Server Cloud [21]

Jährliche Gesamtkosten
Bei einer Nutzung von 1 CPU, 1 GB RAM und 50 GB Speicher und einem Speicherplatz-

[10] Die TÜV-Süd Gruppe verfügt über qualifizierte Experten zu Sicherstellung und Prüfung und Zertifizierung der Arbeitsabläufe.
[11] TELES AG Informationstechnologien: http://www.teles.com/teles.html
[12] Freenet Group: http://www.freenet-group.de/index.html

Starter Paket	1 Cent pro Credit	einmalig 1,99€
Prepaid 10+	2000 Kredite	10€/ Monat
Prepaid 20+	4000 Kredite	20€/ Monat
Prepaid 30+	7500 Kredite	30€/ Monat
Prepaid 40+	15000 Kredite	60€/ Monat

Tabelle 4.7: Strato - Preise der Kreditpakete [21]

bedarf von S1 = 2,73 GB, ist das “Prepaid 10+” Paket ausreichend.

1 CPU	im Monat	=	720 Kredite
1 GB RAM	im Monat	=	720 Kredite
50 GB Speicher	im Monat	=	360 Kredite
Gesamt Kredite	pro Monat	=	1800 Kredite

Das Starter Paket, mit 1 Cent pro Kredit, ist bei einer benötigten Kreditanzahl von 1800 Krediten im Monat, mit 18€ und zusätzlichen Gebühren von 1,99€ höher, gegenüber dem “Prepaid 10+” Tarif.

4.6 Gesamtbetrachtung

Das Kapitel Gesamtbetrachtung umfasst die Kapitel 4.1, Ziel, bis einschließlich Kapitel 4.5, Evaluation Cloud-Systeme.

Für die Betrachtung der Evaluation werden in Form einer Matrix die wichtigsten Punkte der Technologie, Zukunftssicherheit, dem Datenschutz und den Kosten bewertet, siehe Abbildung 4.6. Die Bewertung richtet sich nach der Evaluation der Cloud-Systeme aus Kapitel 4.5. Dabei hat Azure als Anbieter mit 40 Punkten vor Amazon (39 Punkte) und Strato (35 Punkte) abgeschnitten. Die Begründung dazu ist in den nachfolgenden Kapiteln mit der Betrachtung der wichtigsten Punkte beschrieben.

Bewertung (gut=3, befriedigend=2, schlecht=1)

		Azure	Amazon	Strato
Technologie	Infrastruktur	3	2	2
	Framework	3	3	3
	Skalierbarkeit	3	3	3
	Verschlüsselung	3	3	3
	Entwicklerfreundlich	3	2	2
	Verfügbarkeit	3	3	1
Zukunftssicherheit	Marktbeteiligung	2	3	1
	Verfügbar > 5 Jahre	3	3	2
Datenschutz	ISO zertifiziert	3	3	3
	Server Standort	1	2	3
	Kontrollrecht	3	3	3
	Vertraulichkeit	3	3	3
	Integrität	3	3	3
Kosten	laufenden Kosten	2	2	2
	zusätzliche Kosten	2	1	1
Gesamt		40	39	35

Tabelle 4.8: Evaluierung - Matrix

4.6.1 Technologie

Als technologische Eigenschaften sind die Skalierbarkeit des Systems, das verwendbare Framework, die Infrastruktur und die Verschlüsselung von Bedeutung. Die Skalierbarkeit definiert die Möglichkeit zur Erweiterung des Systems, bestehend aus Hardware und Software. Es sollen jederzeit mehr Speicherplatz und weitere Ressourcen nutzbar sein.

Als Cloud-Service-Ebene eignet sich für dieses Projekt "PaaS", siehe Kapitel 3.3.1, am besten, da zum Entschlüsseln und Herunterladen der Bilddaten ein Skript benötigt wird, sowie eine Webseite zum Anzeigen der Bilder im Browser. Zudem ist die Entwicklertechnische Umgebung gegeben. Eine funktionierende "IaaS", siehe Kapitel 3.3.1, muss selbstverständlich als Grundlage vorhanden sein. Das Framework beschreibt den Rahmen der Softwaretechnik, welcher serverseitig zum Einsatz kommen soll. Der Einsatz von Active Sever Pages .NET (ASP.NET) wäre günstig, da SIDEXIS 4 ebenfalls in .NET programmiert (siehe auch Kapitel Grundlagen 3.1) wurde und diese zur Entwicklung von Webanwendungen eine Funktionsvielfalt für Frontend- und Server-Technologien bietet.

Aus dem Gesichtspunkt der Infrastruktur, bietet Azure mit "PaaS" die besten Möglichkeiten, im Gegensatz zu Amazons EC2, welcher eher an IaaS angelehnt ist. Auch Strato mit hoher Skalierbarkeit der Hardwareressourcen ist für die Entwicklung eher ungeeignet. Daher schneidet Azure im Bereich der Infrastruktur in der Auswahl besser ab. .NET, als Framework, wird von Azure, Amazon und Strato unterstützt. Auch die Skalierbarkeit und die Möglichkeit zur Verschlüsselung der Daten ist bei allen Anbietern vorhanden. Bei Azure ist die Entwicklerfreundlichkeit, durch die Möglichkeit Visual-Studio online für die Entwicklung zu integrieren, zu vermerken. Die Verfügbarkeit ist lediglich bei Strato negativ zu bewerten, aufgrund diverser Komplikationen in der Vergangenheit, siehe Kapitel 4.5.3.

4.6.2 Zukunftssicherheit

Um möglichst langfristigen Nutzen zu gewährleisten, ist es erforderlich, dass der Cloud Anbieter nicht in absehbarer Zeit seine Angebote einstellt weil er auf dem Markt nicht länger konkurrenzfähig ist. Mit der strategischen Planung von 5 Jahren ist deshalb die Präsenz des Anbieters am Markt der vergangenen Jahre von Bedeutung. Dies spiegelt sich auch in der Forschung und Innovation, welche der Anbieter betreibt wieder. Ein Unternehmen welches nicht nach neuen Technologien und effizienteren Systemen forscht, scheidet früher oder später durch andere Mitbewerber am Markt aus. [41]

Anhand der Informationen in den Kaptitel Amazon 4.5.2 und Azure 4.5.1 ist die Zukunftssicherheit für diese beiden Anbieter positiv zu bewerten. Besonders Amazon, die mit ihren Web Services bereits 2006 auf den Markt kamen und Microsoft, die 2010 folgten sind in dieser Branche bereits etabliert. Strato hingegen brachte ihr Angebot der Server Cloud erst 2012 auf den Markt. [14] [8] [20]

4.6.3 Datenschutz

Wie in Kapitel 4.2, Datenschutz Einschränkungen, beschrieben wurde, ist ein Cloud Anbieter dann annehmbar, wenn diverse Datenschutzstandards durch Zertifizierungen, wie die der ISO 27001 vorliegen. Ein sicherer Verbindungsaufbau beim Datenaustausch ist ebenso maßgebend wie die Wahrung der Vertraulichkeit und Integrität. Ebenso das Kontrollrecht über die Daten. Des Weiteren ist der Serverstandort in Deutschland kein "Muss"-Kriterium, wie bereits in Kapitel 4.2 festgestellt wurde.

Die Bewertung des Datenschutzes fällt für alle drei Anbieter, bis auf minimale Unterschiede gleich aus. Alle Anbieter garantieren Kontrollrecht der Daten, Vertraulichkeit und Integrität in Form von verschlüsselten Verbindungen und weisen eine annehmbare Zertifizierung aus. Lediglich bei dem Punkt des Serverstandortes gehen die Bewertungen auseinander. Bei Amazon ist einsehbar wo die Daten liegen, allerdings muss dies nicht zwingend in der EU sein. Nur bei Strato ist versichert, dass die Daten nicht außerhalb von Deutschland abgelegt werden. Die Einhaltung des Safe-Harbor-Abkommens auf welches Amazon plädiert ist bedenklich. Denn für dieses Abkommen genügt ein gerichtetes Schreiben an das Handelsministerium der USA, mit der Erklärung, dass man sich den Grundsätzen unterwirft. Eine Überprüfung, ob diese sich auch inhaltlich an die Grundsätze halten findet nicht statt. [14] [8] [20] [47]

4.6.4 Kosten

Bei einer Kostengegenüberstellung der drei Anbieter sind die Kosten mit einer Differenz von weniger als 10€ kaum abweichend, wie in der Abbildung 4.2 verdeutlicht ist.

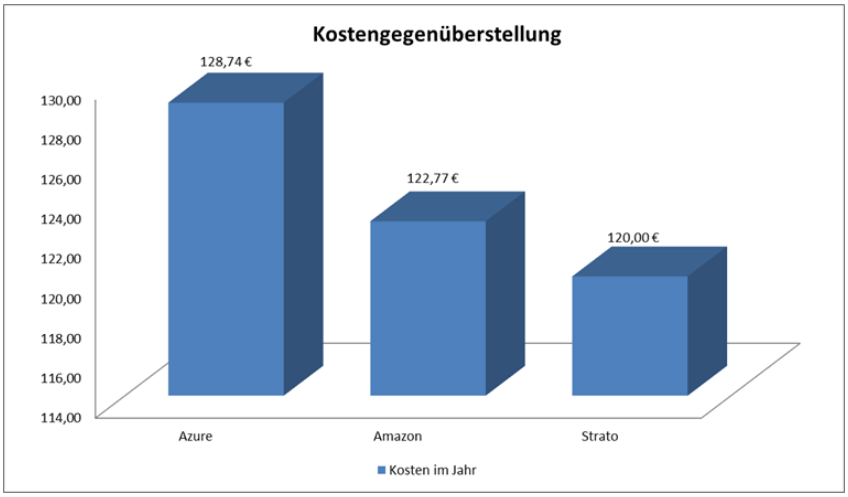


Abbildung 4.2: Kostengegenüberstellung

Ein Kostenbetrag in dieser Höhe ist für Sirona in Bezug auf die Entscheidung des Anbieters nicht ausschlaggebend. Für die laufenden Kosten werden deshalb alle gleich bewertet. Zusätzliche Kosten betreffen bei Strato vor allem einmalige Gebühren zu Beginn der Bereitstellung. Die Kostenberechnung nach "PUT", "LIST", "GET' und weiteren Befehlen, die pro Patienten zum Hochladen und Herunterladen verwendet werden, sind bei Amazon schwerer eingrenzbar als die von Azure, mit reinen Paketpreisen für ausgehende Datenübertragungen.

4.7 Entwurfsentscheidung

Die Gesamtbetrachtung hat ergeben, dass Azure das geeignetste Cloud-System für den zu implementierenden Prototypen ist. Cloud-Speicher und Service werden demnach von Azure bereitgestellt und verwendet.

Dadurch, dass SIDEXIS 4 in der Hochsprache C# und .NET Framework 4.5 programmiert wurde, wird dies ebenfalls für den Prototyp verwendet, der objektorientiert als Modul implementiert wird. Dadurch können bereits vorhandene Klassen zum bereitstellen der Bilddaten verwendet werden.

Die serverseitige Programmierung erfolgt ebenfalls durch C# und ASP.NET, mit Visual-Studio als Entwicklungsumgebung. Somit kann der Code direkt in die Cloud publiziert werden. Die Darstellung erfolgt somit im Rahmen einer Webseite.
Aufgrund der Datenschutz Einschränkungen, werden die Bilder erst nach einer Bestätigung des Patienten in die Cloud hochgeladen. Dies wird anhand eine Bestätigung durch eine Benutzerinteraktion durchgeführt. Außerdem werden die Röntgenbilder pseudonymisiert. Die Verschlüsselung erfolgt durch einen AES 256-Bit Verschlüsselung, da diese zu den sicheren Standard-Verschlüsselungen gehört.

Für die Entfernung der Bilder, soll serverseitig ein Programm einmal täglich ausgeführt werden, welches alle Bilder die älter als 7 tage sind vom Azure-Speicher entfernt. Die Programmierung dieses Programms soll ebenfalls mit der Sprache C# erfolgen.

4.8 Beschreibung des Systems

In diesem Kapitel wird der Aufbau und Ablauf des Systems dargestellt. Außerdem werden differenziert die Komponenten, das Design und die Funktionalität der einzelnen Klassen erklärt. Zudem wird die Schnittstelle genauer betrachtet.

Das System besteht demnach aus der Software SIDEXIS 4 mit dem implementierten Image-CloudModul, der Azure-Cloud mit der ASP.NET-Webseite und dem Speicher, und als Schnittstelle dem Smartphone, welches durch Abfotografieren des dargestellten QR-Codes des Moduls auf die Webseite der Cloud verweist, wie in Abbildung 4.3 veranschaulicht.

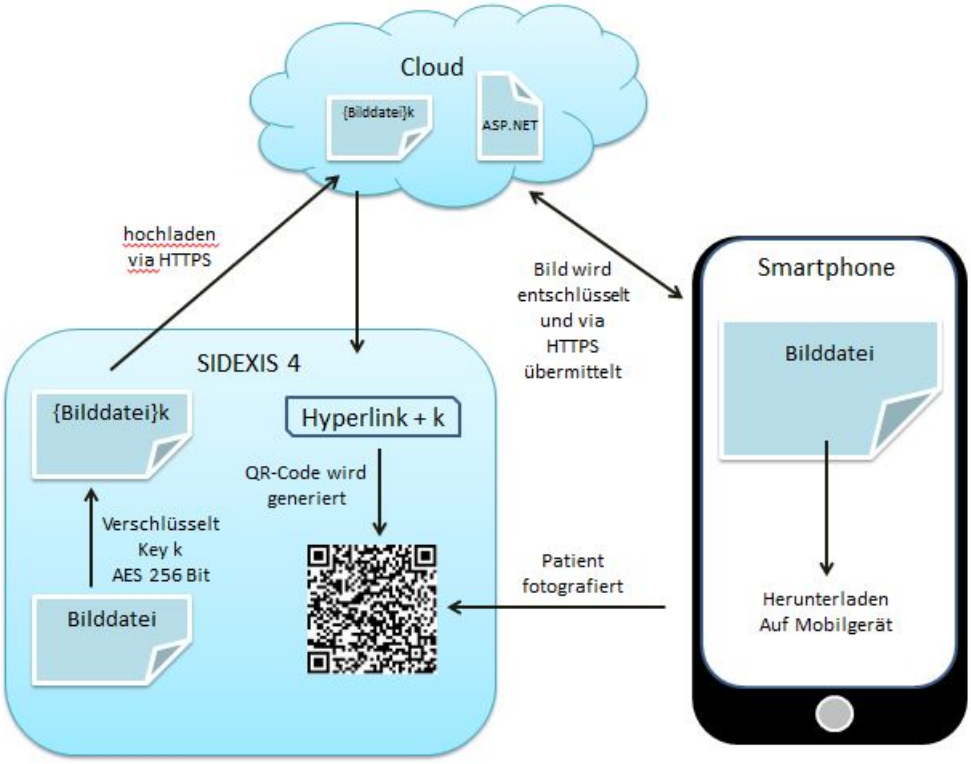


Abbildung 4.3: Entwurf - Beschreibung des Systems

4.8.1 Ablauf

Der Ablauf der Komponenten wird wie folgt beschrieben:

Im Ausgangszustand zum Ausführen des Prototypen befindet sich die Software SIDEXIS 4 in der Phase der Untersuchung, siehe Kapitel 3.1.1, Aufbau der Software, mit der Darstellung der aktuellen Röntgenbildern auf dem Workspace zur Befundung und weiterer Aktionen.

Über eine Schaltfläche zum Auslagern der aktuellen Röntgenbilder, wird das Modul aufgerufen.

Zuerst wird jedes Bild das sich auf dem Workspace befindet in zwei unterschiedliche große Auflösungen gebracht. Diese sind 1024 x 768 Pixel und 1920 x 1080 Pixel. Außerdem werden weitere vorhandene Informationen und Kommentare, wie das Datum, oder der Name des Patienten entfernt. Wurden die Bilder angepasst, werden diese anschließend mit einem generierten Schlüssel 'k' verschlüsselt und in einem lokalen Ordner auf der Festplatte zwischengespeichert.

Nach diesem Vorgang wird in die Phase Ausgabe gewechselt. Dort werden die zuvor verschlüsselten Röntgenbilder angezeigt. Die Darstellung beinhaltet zudem eine Checkbox, welche auf die Auslagerung der Bilddaten in die Cloud hinweist. Sobald diese angewählt wurde, wird eine Schaltfläche unterhalb der Checkbox aktiv. Durch Bestätigen der Schaltfläche werden die zwischengespeicherten Röntgenbilder in einen Container der Cloud hochgeladen. Zu diesem Container in denen sich die Bilddaten befinden wird ein Hyperlink mit dem Schlüssel 'k' als zusätzlichem Parameter erstellt. Dieser Hyperlink wird zu einem QR-Code konvertiert, welcher dann dargestellt wird. Zusätzlich wird eine Verknüpfung zum Standard-Email-Programm dargestellt, sodass der Hyperlink auch per Email an der Patienten gesendet werden kann. Siehe Abbildung 4.6.

Durch Abfotografieren des QR-Codes mit dem Smartphone mittels einer QR-Reader App, wird durch den Hyperlink die ASP.NET Webseite der Cloud aufgerufen. Die Webseite entschlüsselt die relevanten Bilddaten mithilfe der Übergabeparametern des Hyperlinks. Diese werden dann zum Herunterladen in den verschiedenen Auflösungen angeboten. Der Patient erhält somit die Bilddaten in digitalem Format und hochwertiger Auflösung. Zudem kann der Patient seine Bilder jederzeit wieder von der Cloud entfernen.

Ein weiteres serverseitiges Skript ist für die Entfernung der Bilder zuständig und wird täglich ausgeführt. Durch dieses Skript werden alle Bilder die älter als 7 Tage sind automatisch von der Cloud entfernt.

4.8.2 SIDEXIS 4 Modul Design

Das Modul wird in verschiedene Klassen implementiert und umfasst die Bereitstellung und Anpassungen der Bilddaten, deren Verschlüsselung, dass Hochladen, die Generierung des QR-Codes, sowie die Darstellung mit der Benutzerinteraktion des Patienten.

Steuerung

Die "ImageCloudModel"-Klasse steuert den gesamten Ablauf des Moduls und enthält die Business-Logik. Sie erhält über ein Event der "ImageCloudPrintAction" -Klasse die Bilder in hoher und niedriger Qualität, sowie die Seite zum Darstellen der Bilder. Die "ImageCloud-Model" -Klasse verwendet verschiedene Hilfsklassen zum verschlüsseln der Bilder, Erzeugen des QR-Codes und Hochladen der Bilder in den Azure-Speicher. Sie setzt Eigenschaften, die über eine "OnPropertyChange" -Methode an die "ImageCloudViewModel" -Klasse gemeldet werden, wie in Abbildung 4.5 veranschaulicht.

Bildbereitstellung

Die Klasse "ImageCloudPrintAction" wird über den Button der "ImageCloudWidget" -Klasse aufgerufen. Diese entfernt von den Bildern des aktuellen Workspaces die Patienteninformationen, wie den Namen und die Uhrzeit der Aufnahme. Die Bilder werden in eine hohe und eine niedrige Auflösung gerendert und jeweils in Bitmap-Source-Listen gespeichert. Durch die Verwendung einer Klasse, die zum Drucken der aktuellen Darstellung des Workspaces dient, entnehmen wir die Page zum Darstellen der Bilder, welche auf die Cloud hochgeladen werden sollen.

Die Eigenschaften "*Document*", sowie die beiden Bitmap-Source-Listen der "ImageCloud-WorkflowResult" -Klasse, werden als Objekt über ein Event an die Methode "OnImage-CloudPreview" der "ImageCloudMode" -Klasse übergeben.

Verschlüsselung der Bilddaten

Die Klasse "GenerateNameKeyEncryptionHelper" ist für die Generierung der Schlüssel für die Verschlüsselung und Generierung der Namen, mit denen die Bilder auf der Cloud abgespeichert und zugeordnet werden können, zuständig. Dazu dient eine interne "Random-String" -Methode die als Übergabeparameter die Länge des zurückzugebenden Zufall-Strings definiert.

Auch der Verschlüsselungsprozess wird durch die Öffentliche Methode "AES_Encrypt()" in dieser Klasse vorgenommen. Diese benötigt als Übergabeparameter die Bitmap, einen String der definiert ob das Bild eine hohe oder niedrige Auflösung hat und den lokalen Speicherplatz, wo die Bilder nach der Verschlüsselung zwischengespeichert werden. Zur Verschlüsselung wird die "RijndaelManaged" -Klasse und die "Rfc2898DeriveBytes" -Klasse verwendet.

Benutzeroberfläche und Interaktion

Die "ImageCloudView"-Klasse ist für die Benutzeroberfläche zuständig. Die Darstellung erfolgt mithilfe des Grafik-Frameworks Windows Presentation Foundation (WPF) welches Be-

standteil des .NET Frameworks ist. Sie stellt den QR-Code, sowie die Röntgenbilder dar, die hochgeladen werden sollen und erhält die verschiedenen Eigenschaften zum Einbinden über die ”ImageCloudViewModel” -Klasse. Außerdem werden die Benutzerinteraktionen durch checken der Checkbox und bestätigen der Button abgefangen und ausgewertet.

Die Benutzerinteraktion ist so konzipiert, dass bei einem Patienten, der die Datenschutzerklärung nicht akzeptiert, keine Bilder in den Cloud-Speicher geladen werden. Sobald die Checkbox aktiviert wird, kann auch der Button zum hochladen der Bilder betätigt werden. Nach betätigen des Hochlade-Buttons, werden die Checkbox und der Button deaktiviert um das doppelte Hochladen der Bilder zu vermeiden. Als ausbleibende Benutzerinteraktion kann der Patient die View wieder verlassen. Dazu dienen die beiden Methoden ”OnApply()”, welche die Methode ”UploadImages()” der ”ImageCloudModel” -Klasse aufruft, wenn der Button zum Hochladen aktiviert wurde, und ”OnBack()”, welche die ”Finish()” -Methode der Klasse ”ImageCloudModel” aufruft, um das Model ”aufzuräumen”, sobald die View wieder verlassen wird.

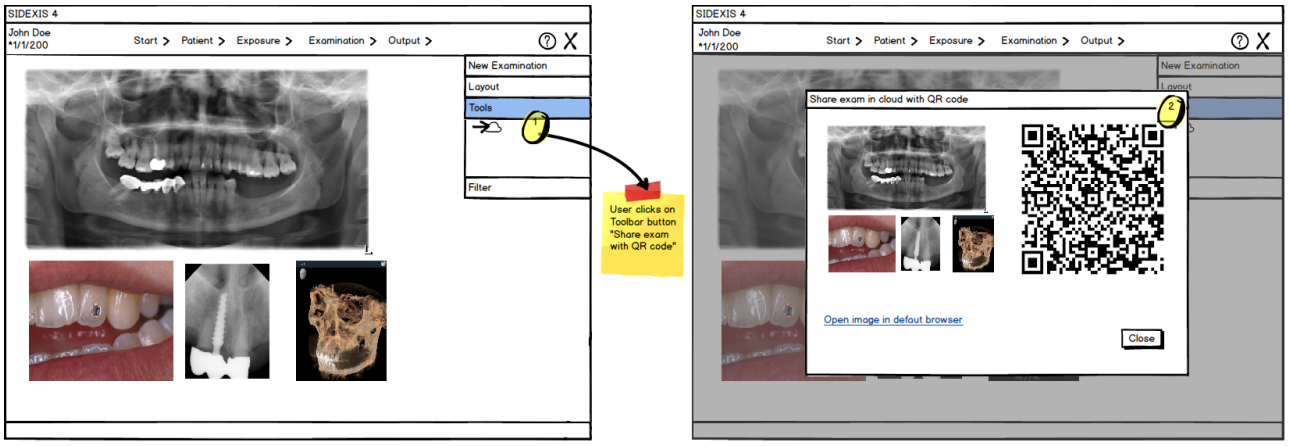


Abbildung 4.4: Konzept der Benutzeroberfläche von Bernhard Schmitt

Generierung des QR-Codes

Die Klasse ”QRCodeEncoder” erzeugt mit der ”Encode()” -Methode aus einem String einen Zweidimensionalen QR-Code und gibt diesen in Form einer Bitmap zurück. In diesem Fall enthält der String den Verweis auf die Cloud-Webseite und enthält diverse Übergabeparameter. Die Klasse ist ein öffentliches Paket und Teil des .NET Frameworks und kann über Visual-Studio heruntergeladen und verwendet werden.

Hochladen der Bilder

Die Klasse "UploadPicturesHelper" stellt mit den benötigten Account-Infos eine autorisierte Verbindung zum Azure-Speicher her und lädt die Bitmaps aus einem Ordner in einen Container. Die Klasse bietet dafür eine Methode "UploadPictures" die von der Methode "UploadImages" der "ImageCloudModel" -Klasse aufgerufen wird und benötigt zwei Strings als Übergabeparameter. Der "*container*" definiert den Container im Cloud-Speicher, in dem die Bilder gespeichert werden sollen und der "$localFolderPath$" definiert den Speicherpfad, in dem sich die verschlüsselten Bilder zum Hochladen befinden.

Abhängigkeit der Klassen

Die Abhängigkeit der Klassen die zuvor im einzelnen beschrieben wurden, wird in Abbildung 4.5, mit den verschiedenen Beziehungen veranschaulicht.

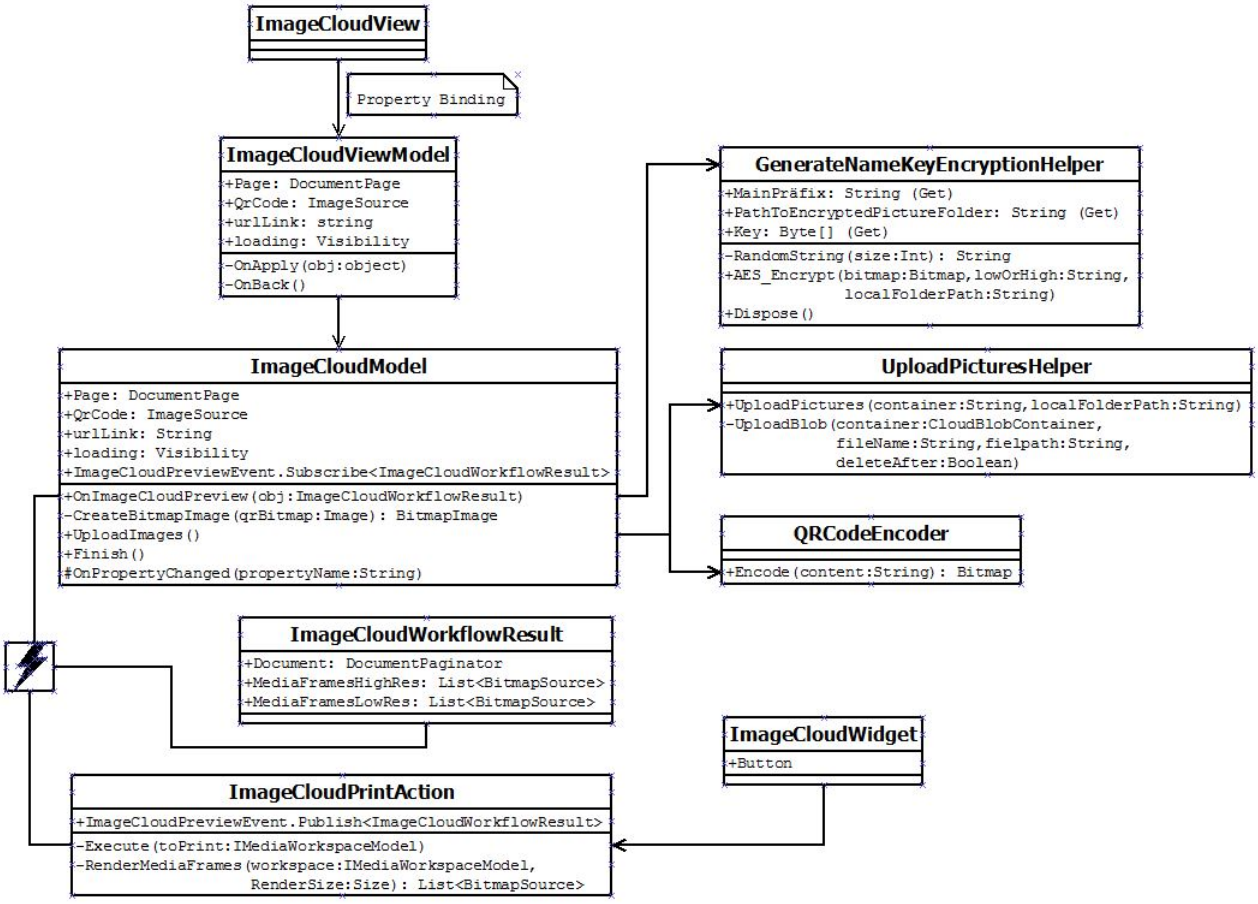


Abbildung 4.5: SIDEXIS 4 Modul - Klassendiagramm

Systemverhalten

Wie bereits im Ablauf, in Kapitel 4.8.1, darauf eingegangen wurde, ist in Abbildung 4.6 das Verhalten des SIDEXIS 4 Moduls mithilfe eines Flussdiagrammes veranschaulicht.

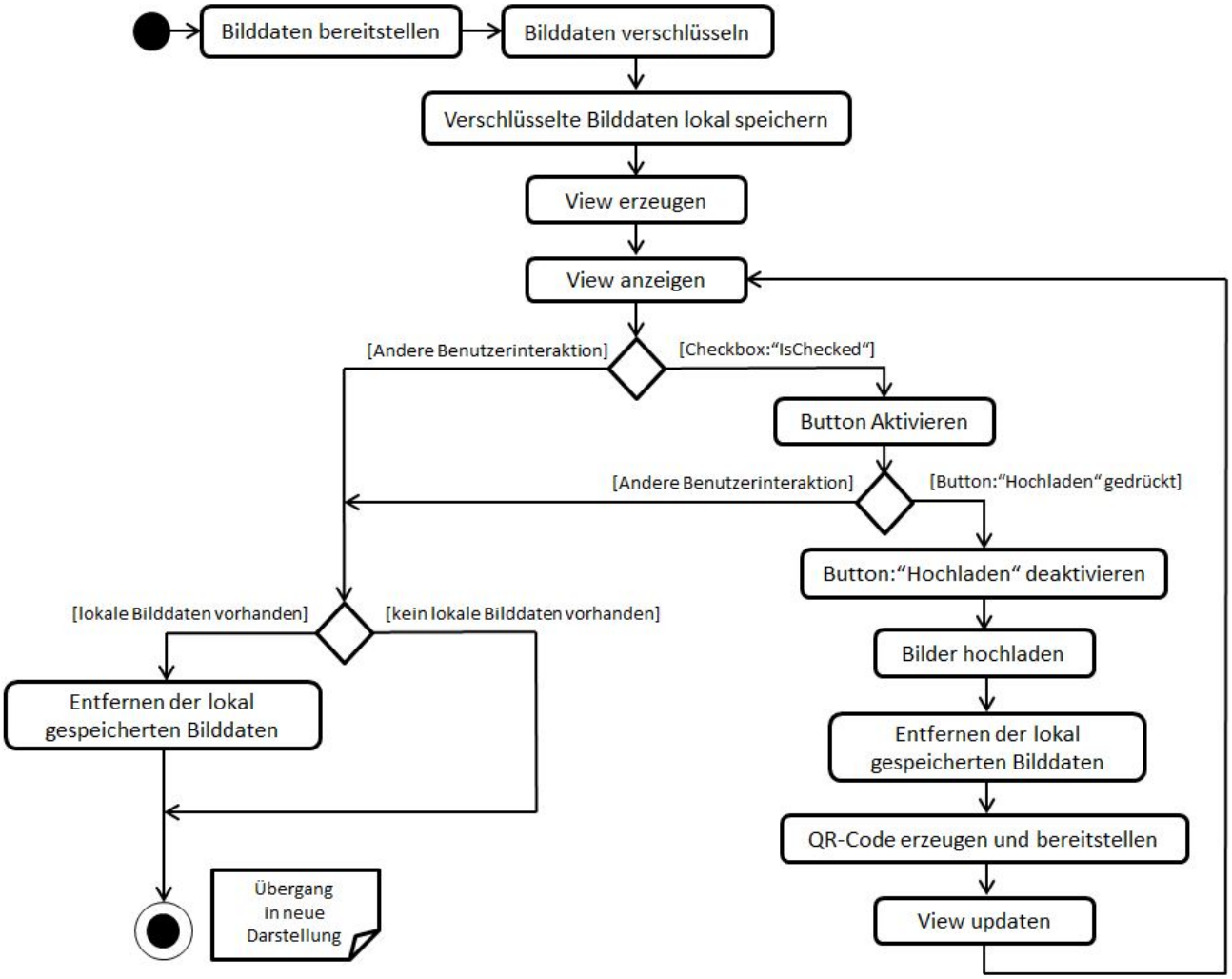


Abbildung 4.6: Systemverhalten - Flussdiagramm des Moduls

4.8.3 Azure Cloud Design

Die Azure Cloud besteht aus einer ASP.NET Webseite mit einer Klassse HomeController als Controller, die eine Aktion für jede View bereitstellt. Eine Aktion ist in Form einer Methode implementiert. Die Aktionen welche die HTTP-Anfragen entgegennehmen, werden als ”Public” deklariert. Alle restlichen Funktionen und Methoden müssen als ”Private” deklariert werden.

Des weiteren wird ein Programm zur automatischen Entfernung der Bilder im Cloud-Speicher einmal am Tag ausgeführt

Startseite

Als Hauptseite der Webseite, dient die Aktion-Methode "Index". Auf dieser finden sich Informationen zum Datenschutz des Projektes, sowie ein Verweis auf die Sirona-Produktseite. Bei falschen Anfragen an den Controller wird automatisch auf die "Index" -View weitergeleitet.

Bild-Zuweisung

Die Aktion-Methode "FindImage" ist für die Entgegennahme der HTTP-Anfrage von dem verweisenden Hyperlink des QR-Codes zuständig. Dafür definiert die Klasse "RouteConfig", wie genau die URL verarbeitet wird und welche Werte als Übergabeparameter zählen.
Diese werden dann auf ihre Richtigkeit überprüft, um anschließend eine Liste aller vorhandenen Bilder zu diesen Parametern zusammen zu stellen.

Die dazugehörige View als HTML-Seite stellt die vorhandenen Bilder des Verweises als Vorschau dar und bietet das Herunterladen und löschen der Bilder an.

Verbindung zum Cloud-Speicher

Die Verbindung und Referenzierung zu den Containern des Cloud-Speichers wird mithilfe der "BlobStorageServices" -Klasse realisiert. Zur Referenzierung eines Containers des Azure-Speichers, wird ein Name des Containers vom Datentyp String als Übergabeparameter erwartet.

Qualität der Bilder

Für die Unterscheidung und Zuweisung der richtigen Auflösung beim Herunterladen der Bilder, stehen zwei Aktion-Methoden zur Verfügung. Diese sind "BildHigh" und "BildLow" und bestimmen, welches Bild von der Azure-Cloud geladen werden soll.

Herunterladen der Bilder

Die "LoadPic" -Methode ist für das Herunterladen der Bilder zuständig. Sie erwartet den Containername in dem sich die Bilder befinden, den Namen des Bildes, sowie den Schlüssel zur Verschlüsselung. Damit stellt sie mithilfe der "BlobStorageServices" -Klasse die Verbindung zum Azure-Speicher her und lädt die verschlüsselten Bilder in ein Byte-Array. Dieses

wird anschließend wieder zurückgegeben.

Entschlüsselung der Bilder

Für die Entschlüsselung sorgt die "AES_Decrypt" -Methode. Diese erhält ein Byte-Array mit dem verschlüsselten Bild, sowie ein Byte-Array mit dem Schlüssel und gibt das Bild als entschlüsseltes Byte-Array zurück. Die "AES_Decrypt" -Methode wird von der "LoadPic" -Methode aufgerufen.

Manuelle Entfernung der Bilder

Die Aktion-Methode "Delete" ist für das Löschen der Bilder zuständig. Diese ermöglicht dem Patienten über die Webseite die manuelle Löschung von einzelnen Bildern des Verweises, oder allen Bildern. Als Übergabeparameter wird ein String welcher den Namen des Bildes, den Schlüssel für die Entschlüsselung und ein Boolean erwartet. Der Boolean definiert, ob beide Auflösungen eines einzelnen Bildes gelöscht werden soll, oder alle Bilder die sich auf den selben Präfix des aktuellen Verweises beziehen.

Nach der Löschung der Bilder wird eine weitere Aktion-Methode "Confirmation" mit der dazugehörigen View aufgerufen, welche den erfolgreichen Vorgang der Entfernung der Bilder auf dem Azure-Speicher bestätigt.

Automatische Entfernung der Bilder

Für die Automatische Entfernung der Bilder dient ein Programm "CleanUp". Dieses Programm entfernt alle Bilder die älter als 7 Tage sind mithilfe der Klasse "BlobStorageService", die zur Verbindung und Authentifizierung des Azure-Speichers dient.

4.8.4 Schnittstelle Azure - SIDEXIS 4

Die Schnittstelle, zum Verbinden des Moduls in SIDEXIS 4 und der ASP.NET Weibseite der Azure-Cloud, ist ein Hyperlink, welcher als QR-Code dargestellt wird. Der Hyperlink besteht aus der Adresse der Azure-Cloud-Webseite, dem zu verwendenden Controller, der Aktion-Methode, sowie den Übergabeparametern. Die Übergabeparameter beinhalten den Namen des Containers auf dem Azure-Speicher, den Präfix als Teil des Namens der Bilder und den Schlüssel zur Entschlüsselung der Bilder:

$$\underbrace{http://pasilgtestcloud.cloudapp.net}_{\text{AzureCloud}}\underbrace{/Home}_{\text{Controller}}\underbrace{/FindImage}_{\text{Aktion-Methode}}\underbrace{/2015-08-05}_{\text{Container}}\underbrace{/5alvh}_{\text{Präfix}}\underbrace{/3anFi44S}_{\text{Schlüssel}}$$

Container

Jedes Bild wird in einen Container auf dem Azure-Speicher geladen. Dabei wird überprüft, ob der Container bereits existiert. Existiert dieser nicht, wird ein neuer angelegt. Der Name des Containers ist immer das Aktuelle Datum in dem Format: "$yyyy - MM - dd$".

Präfix

Werden mehrere Bilder pro Patient hochgeladen, ist die Namenskonvention der Bilder von Bedeutung. Hierbei bekommen alle Bilder im vorderen Teil des Namens einen identischen Präfix. Somit kann die Aktion-Methode "FindImage" der Azure-Cloud entsprechend dem Verweis des QR-Codes alle Bilder referenzieren. Der Präfix wird automatisch generiert.

Schlüssel

Der Schlüssel dient zur Entschlüsselung der Bilddaten. Dieser wird wie der Präfix für einen Vorgang des Hochladens von x Bildern generiert und auf diese angewendet. Die Übergabe des Schlüssels ermöglicht das Entschlüsseln aller Bilder eines QR-Code-Verweises auf der Azure-Cloud.

5 Implementierung

In diesem Kapitel wird die Programmierung der AES-Verschlüsselung und dem Hochladen der Bilddaten des Moduls aufgezeigt.

Für die serverseitige Programmierung der Azure Cloud erfolgt eine Erläuterung über das Verweisen der Bilder, dem Herunterladen und der automatischen Bildbereinigung.

5.1 SIDEXIS 4 Modul

In diesem Kapitel wird die Implementierungen der AES-Verschlüsselung, dem Hochladen der Bilder in den Azure-Speicher und der Benutzeroberfläche des SIDEXIS 4 Moduls erklärt.

5.1.1 AES-Verschlüsselung

Die Funktion AES-Encrypt der Klasse "*GenerateNameKeyEncryptionHelper*" erhält die zu verschlüsselnde Bitmap, einen String "*lowOrHigh*" und den Speicherpfad. Der String "*lowOrHigh*"gibt an, ob das Bild eine hohe oder niedrige Auflösung hat. Der Speicherpfad definiert den Pfad unter welchem das verschlüsselt Bild zwischengespeichert werden soll.

Als Erstes wird mithilfe einer internen Methode "setSavePathFile" der Speicherpfad mit vollständigem Name des Bildes in einer String-Variable "*savepathFile*" erstellt. Der Name setzt sich dabei aus einem "*MainPräfix*", "*SubPräfix*" und der JPEG-Bild-Format-Endung zusammen. "*MainPräfix*" und "*SubPräfix*" werden mit einem "_" unterteilt und mithilfe der "*randomString*()" -Methode bereitgestellt. Der "*MainPräfix*" besteht aus 5 Zufallszeichen und gibt an, welche Bilder zu einem Verweis eines erzeugten QR-Codes gehören. Der "*SubPräfix*" beginnt entweder mit "*low*" für Bilder mit niedrigerer Auflösung, oder "*high*" für Bilder mit hoher Auflösung und endet mit weiteren 4 Zufallszeichen. Unter der String-Variable "*savepathFile*" wird die Bitmap dann abgespeichert und über den Aufruf "*File.ReadAllBytes*()" in ein Byte-Array gelesen.

Für das Verschlüsseln der Bilder wird die Klasse RijndaelManaged verwendet. Diese gehört dem Namespace System.Security.Cryptography an, welcher für die Bereitstellung der Kryptografiedienste zur Verfügung steht.
Somit wird ein neues RijndaelManaged Objekt erzeugt, welches in Listing 5.1 mit "*AES*" gekennzeichnet ist. Die Schlüsselgröße wird auf 256 Bit und die Blockgröße auf 128 Bit festgelegt. Der "Key", der ebenfalls durch den Konstruktor mithilfe der "*randomString*()

-Methode und anschließendem hashen erzeugt wurde, wird nicht direkt zur Verschlüsselung verwendet. Dieser wird zusammen mit einem "*Salt*" und einer "*Iteration*" an die Rfc2898DeriveBytes -Klasse übergeben, durch die eine neue Instanz erzeugt wird, die wiederum zum Ableiten des eigentlichen Schlüssel und Initialisierungsvektors dient. Der "*Salt*" ist ein Byte-Array welcher größer als 8 Bytes sein muss. Hier wird er mit 16 Byte deklariert. Die "*Iteration*" legt die Schritte zur Wiederholung der Operation fest welche auf 1000 gesetzt wird. Beide Werte müssen nicht geheim sein.
Die Ableitung des Schlüssels und des Initialisierungsvektors wird durch das Objekt mithilfe einer "*GetBytes*()" Methode realisiert. Da die Schlüsselgröße auf 256 Bit definiert wurde, muss bei dieser Anweisung die Schlüsselgröße durch 8 dividiert werden, da Bytes von der Funktion zurückgegeben werden. Dasselbe gilt für die Blockgröße. Zuletzt ist der Verschlüsselungsmodus auf "*CBC*" zu setzen.

Die Instanz "*CryptoStream*" verknüpft einen Stream zu der kryptografischen Transformation. Wie in Listing 5.1, wird dazu ein Memory-Stream verwendet. Nachdem die Transformation abgeschlossen ist, wird das unverschlüsselte Bild unter dem Speicherpfad "*savepathFile*" entfernt, und durch das verschlüsselte Bild das von dem Memory-Stream zu einem Byte-Array konvertiert wird unter dem selben Namen gespeichert.

```
public static void AES_Encrypt(Bitmap bitmap, string lowOrHigh, string
    localFolderPath)
{
   var savepathFile = setSavePathFile(localFolderPath, lowOrHigh, Resources.JPG);
   bitmap.Save(savepathFile);
   byte[] bytesToBeEncrypted = File.ReadAllBytes(savepathFile);

   using (var ms = new MemoryStream())
   {
     using (RijndaelManaged AES = new RijndaelManaged())
     {
       AES.KeySize = 256;
       AES.BlockSize = 128;

       byte[] saltBytes = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
           };
       var key = new Rfc2898DeriveBytes(passwordBytes, saltBytes, 1000);

       AES.Key = key.GetBytes(AES.KeySize / 8);
       AES.IV = key.GetBytes(AES.BlockSize / 8);
       AES.Mode = CipherMode.CBC;

       using (var cs = new CryptoStream(ms, AES.CreateEncryptor(),
           CryptoStreamMode.Write))
       {
         cs.Write(bytesToBeEncrypted, 0, bytesToBeEncrypted.Length);
         cs.Close();
       }
       File.Delete(savepathFile);
       File.WriteAllBytes(SavepathFile, ms.ToArray());
     }
   }
}
```

Listing 5.1: AES-Verschlüsselung

Die Entschlüsselung, welche serverseitig stattfindet, erfolgt auf die selbe Weise. Der einzige Unterschied zu dem Code in Listing 5.1 ist die Verwendung der "*CreateDecryptor*()" -Methode bei dem Aufruf des "*CryptoStreams*" des Rijndael-Objekts.

5.1.2 Hochladen der Bilder

Für das Hochladen der Bilddaten in der "*UploadPicturesHelper*"-Klasse, wird eine Verbindung mit dem Azure-Cloud-Speicher hergestellt. Dazu wird die öffentliche Bibliothek Windows Azure Storage von Microsoft verwendet. Diese erlaubt das Verwalten der Speichercontainer, sowie den Datenaustausch zwischen dem Client und dem Azure-Speicher.

Bei dem Aufruf für das Hochladen der Bilder, der in der "ImageCloudModel" -Klasse geschieht, ist es wichtig, dass dieser in einem neuen Thread platziert wird. Dadurch wird sichergestellt, dass die Benutzeroberfläche nicht einfriert, sondern weiterhin während der Zeit des Hochladens der Bilder für den Benutzer verfügbar ist. Dieser Aufruf wird wie in Listing 5.2 implementiert:

```
var task = Task.Factory.StartNew(() =>
{
        UploadBitmap.Laden(container, localFolderPath);
}, TaskCreationOptions.None);
```

Listing 5.2: Thread-Aufruf der Upload Methode

Die Eigentliche Methode zum Hochladen der Bilder erhält als Übergabeparameter den "*container*" und "*localFolderPath*". Der "*container*" definiert den Container in den die Bilder auf dem Cloud-Speicher gespeichert werden sollen. Der "*localFolderPath*" bestimmt den Ordnerpfad zu den verschlüsselten Bildern auf der Festplatte.

Für eine sichere Verbindung, wird der "*ConnectionString*" benötigt. Dieser ist in der SIDEXIS 4 - Konfigurations-Datei hinterlegt und beinhaltet das Protokoll, den Account-Name und den Account-Key, wie in Listing 5.3 ersichtlich ist.

```
<connectionStrings>
  <add name="AzureStorageAccount"
  connectionString="DefaultEndpointsProtocol=https;AccountName=testname;
  AccountKey=testkey"/>
</connectionStrings>
```

Listing 5.3: Connection-String zur Cloud-Authentifizierung

Mithilfe des ConnectionStrings, findet die Referenzierung zum Cloud-Speicher-Account statt und ermöglicht wiederum die Erzeugung eines "*BlobClients*', der dann auf die Container referenziert werden kann. Ist der Container noch nicht vorhanden, wird ein neuer erzeugt.

Wie in Listing 5.4 implementiert, wird der Ordner mit den verschlüsselten Bildern mithilfe einer "*foreach*()" Schleife durchlaufen und durch eine weitere Funktion "*UploadBlob*" hochgeladen.

Diese erhält als Übergabeparameter den referenzierten Container, den Namen des Bildes und den Speicherpfad zu dem Bild. Durch die Anweisung "*container.GetBlockReference*((*file*−*Name*))" wird ein neuer Block referenziert. Über diesen kann mithilfe eines File-Streams das Bild in den Azure-Speicher geladen werden. Im Anschluss wird das verschlüsselte Bild auf der lokal gespeicherten Festplatte durch die Anweisung "*File.Delete*()" gelöscht. Somit werden die Bilder nacheinander hochgeladen.

```
public static void Laden(string blobFolder, string localFolderPath)
{
   string connString = ConfigurationManager.ConnectionStrings
                                  ["AzureStorageAccount"].ConnectionString;

   CloudStorageAccount storageAccount = CloudStorageAccount.Parse(connString);
   CloudBlobClient blobClient = storageAccount.CreateCloudBlobClient();
   CloudBlobContainer container = blobClient.GetContainerReference(blobFolder);
   container.CreateIfNotExists();

   string[] fileEntries = Directory.GetFiles(localFolderPath);
   foreach (var filepath in fileEntries)
   {
     var fileName = Path.GetFileName(filepath);
     UploadBlob(container, fileName, filepath);
     File.Delete(filepath);
   }
}


private static void UploadBlob(CloudBlobContainer container, string fileName,
    string filepath)
{
   CloudBlockBlob block = container.GetBlockBlobReference((fileName));

   using (var fs = System.IO.File.Open(filepath, FileMode.Open, FileAccess.Read,
       FileShare.None))
   {
     block.UploadFromStream(fs);
   }
}
```

Listing 5.4: Connection-String - Azure-Storage-Account

5.1.3 Benutzeroberfläche

Die "*ImageCloudView*" realisiert die Benutzeroberfläche, die mithilfe eines Grid's unterteilt wird. Somit gibt es 3 Reihen. Die oberste ist für die Überschrift gedacht, sodass der Benutzer feststellen kann, wo er sich befindet. Die Überschrift wird in einem Text-Block dargestellt, wobei der Text auf die Resourcen-Datei verweist in welcher sich der Text befindet.
Die zweite Reihe macht den größten Bereich aus. Hier wird die Auswahl der Bilder angezeigt die in die Cloud geladen werden sollen. Die Bilder in der "*Page*" vom Typ "*Document* − *Page.Visual*" werden von der Klasse "*ImageCloudViewModel*" als Eigenschaft bereitge-

stellt und mithilfe eines "$Conten - Presenters$" in die Benutzeroberfläche eingebunden, wie in Listing 5.5 ersichtlich ist.

```
<ContentPresenter Grid.Column="0" Content="{Binding Page}" />
```

Listing 5.5: Benutzeroberfläche - Binding der Bilder

Auf der linken unteren Seite befindet sich ein Button, der zurück zur Phase Untersuchung führt (zu den Phasen von SIDEXIS siehe Kapitel 3.1.1). Das "Command-Attribut" bindet eine Eigenschaft "$BackCommand$", dass von der Klasse "$ImageCloudViewModel$" erzeugt und abgefangen wird. Über das "Style-Attribut" erhält der Button das für SIDEXIS 4 vorgesehenen Design.

Auf der Linken Seite wird der QR-Code zum Abfotografieren mithilfe eines weiteren "Content-Presenters" eingebunden. Da der QR-Code in Form einer "Image-Source" von der Klasse "$ImageCloudViewModel$" bereitgestellt wird, kann dieser auch als Image eingebunden werden. Der QR-Code wird erst dann angezeigt, wenn dieser auch erzeugt wurde. Die Länge und Breite wird auf 200 x 200 Pixel gesetzt. Außerdem ist es notwendig einen weißen Rahmen um den QR-Code zu legen, da es sonst, auf Grund des dunklen Hintergrundes, zu Problemen beim Erkennen und Abfotografieren des QR-Codes führen kann.

Des weiteren befindet sich auf dem "Image-Element" des QR-Codes ein weiterer transparenter Rahmen der über "$Controls : ProcessIndicator$" einen Lade-Zyklus darstellt. Dieser wird Sichtbar für die Zeitspanne vom Beginn des Hochladens der Bilder, bis der QR-Code generiert wurde und angezeigt werden kann. Dies legt das "Visibility-Attribut" fest, welches eine Eigenschaft "$loading$" vom Typ Visibility der Klasse "$ImageCloudViewModel$" einbindet.

Auf der rechten unteren Hälfte befindet sich eine "Checkbox", wie in Listing 5.6 dargestellt, mit dem Namen "DisclaimerAcceptedCB" und weist den Patienten darauf hin, dass dieser sich durch checken der Checkbox mit der Auslagerung der Bilder in die Cloud einverstanden erklärt. Dieser wird über das "Content-Attribut" angezeigt und verweist auf die Ressourcen. Die Checkbox ist beim erzeugen der View standardmäßig nicht gecheckt.

```
<CheckBox
  Name="DisclaimerAcceptedCB"
  Grid.Row="0" Margin="20,50,20,10"
  FontSize="14"
  Content="{x:Static p:Resources.DISCLAIMER_Checkbox}"
/>
```

Listing 5.6: Benutzeroberfläche - Checkbox

Unter der Checkbox befindet sich ein Hyperlink mit dem Name "emailLink" der es ermöglicht, anstatt dem Abfotografieren des QR-Codes, den Verweis als Link zu den hoch geladenen Bildern als E-Mail zu versenden. Das "NavigateUri-Attribut", wie in Listing 5.7 ersichtlich, bindet eine weitere Eigenschaft "$urlLink$" der Klasse "$ImageCloudViewModel$" ein, welches den Link enthält. Der Hyperlink wird erst dann aktiv, wenn der Button zum Start des

Hochladens der Bilder aktiviert wurde. Bei aktivieren des Links, wird eine neue E-Mail mit dem Standard-E-Mail-Programm geöffnet und der Link eingefügt. Der Text dazu, der über das "Text-Attribut" angezeigt wird auf eine Text-Ressource referenziert.

```
<TextBlock Grid.Row="1" Margin="20,10,20,30" FontSize="14">
  <Hyperlink
     Name="emailLink"
     NavigateUri="{Binding urlLink}"
     RequestNavigate="OnNavigate"
     IsEnabled="false">
    <TextBlock Text="{x:Static p:Resources.EMAIL_Hyperlink}"/>
  </Hyperlink>
</TextBlock>
```

Listing 5.7: Benutzeroberfläche - E-Mail link

Unter dem Hyperlink befindet sich ein Button der das Hochladen der Bilder in die Cloud startet. Dieser kann nur dann aktiviert werden, wenn die Checkbox darüber gecheckt wurde. Dies geschieht über das "IsEnabled-Attribut", welches die Checkbox über den Name des Elements und dessen Status "IsChecked" bindet, wie durch Listing 5.8 nachvollziehbar ist. Wird der Button aktiviert, wird über das "Command-Attribut" eine Eigenschaft "$Apply-Command$" der Klasse "$ImageCloudViewModel$" aufgerufen, um das Hochladen der Bilder, sowie der Generierung des QR-Codes zu starten. Wie in Listing 5.9 ersichtlich, ist diese Aktion nur einmal möglich, da nach Betätigung des Buttons die Checkbox, sowie der Button deaktiviert werden, durch setzen des "$IsEnabled$" auf "$false$". Zusätzlich wird der Hyperlink aktiv gesetzt.

```
<Button Command="{Binding ApplyCommand}"
        Name="Uploadbtn"
        Click="ButtonBase_OnClick"
        IsEnabled="{Binding ElementName=DisclaimerAcceptedCB, Path=IsChecked}"
        Grid.Row="2"
        Margin="20,20,20,20" \\
        Width="140"
        Height="40"
        HorizontalAlignment="Left"
        VerticalAlignment="Bottom"
        Content="{x:Static p:Resources.UPLOAD_Button}"
/>
```

Listing 5.8: Benutzeroberfläche - Hochladen Button

```
private void ButtonBase_OnClick(object sender, RoutedEventArgs e)
{
   DisclaimerAcceptedCB.IsEnabled = false;
   Uploadbtn.IsEnabled = false;
   emailLink.IsEnabled = true;
}
```

Listing 5.9: Benutzeroberfläche - Hochladen Button

Den Hauptbereich der Benutzeroberfläche macht die Darstellung der Bilder aus. Mithilfe einer Scrollbar in der Vertikalen, können auch bei mehreren Bildern alle eingesehen werden. In Abbildung 5.1 wird eine Übersicht der beschriebenen Benutzeroberfläche veranschaulicht:

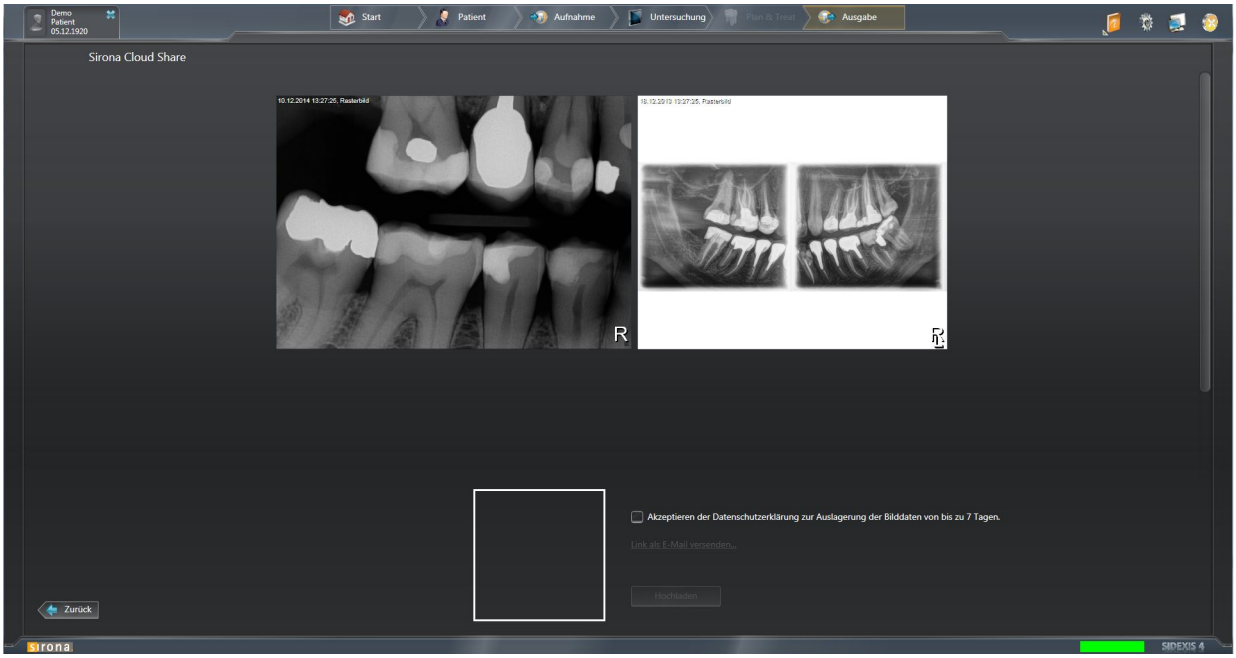


Abbildung 5.1: Benutzeroberfläche - Screenshot der Darstellung

5.2 Azure Cloud

In diesem Kapitel werden Teile der Implementierung der Azure-Cloud genauer erklärt. Dies beinhaltet die Verweise zu den Bildern durch den QR-Code und die Automatische Bildbereinigung.

5.2.1 Verweisen der Bilder

Der Hyperlink der als Verweis auf die Bilder in Form eines QR-Codes dargestellt wird, ruft die Aktion-Methode "*FindImage*" auf und erwartet drei Strings als Übergabeparameter. Diese sind der Name des Containers, der Präfix, welcher die Bilder zu einem Verweis definiert und der Schlüssel zur Entschlüsselung.

Zunächst wird eine String-Liste, "*strListe*" siehe Listing 5.10, angelegt und überprüft, ob einer dieser drei Werte "*null*" ist. Ist dem so, wird die leere String-Liste an das Model übergeben und die View aufgerufen. Durch einen Verweis eines QR-Codes sind diese 3 Werte immer gesetzt und vorhanden. Allerdings kann durch die Manuelle Eingabe eines Nutzers ein Parameter vergessen werden oder ähnliches. Deshalb werden zu Anfang die 3 Parameter sicherheitshalber überprüft.
Sind alle drei Parameter vorhanden, wird mit Hilfe der "*BlobStorageServices*" -Klasse auf den Container referenziert und eine Liste aller Elemente in dem Container befüllt. Sind keine Elemente in der Liste vorhanden, wird der Container auf dem Azure-Speicher gelöscht und wiederum die leere String-Liste "*strListe*" an das Model übergeben und die View aufgerufen.

Andernfalls wird mithilfe einer "*foreach*()" -Schleife jedes Element in der "*blobs*" -Liste mit dem Präfix verglichen. Da ein Element den gesamten Speicherpfad der Cloud mit Dateiname enthält, muss der eigentliche Dateiname zuerst noch abgesondert werden. Durch die einheitliche Namenskonvention ist die Länge der Dateinamen bekannt. Dieser besteht aus einem 5 Zeichen langen "MainPräfix", einem Zeichen "_" zur Unterteilung, 8 weiteren Zeichen als "SubPräfix" und der Dateiendung ".*jpg*".
Durch die "*Substring*()" Anweisung, können die letzten 18 Zeichen gezielt in einen neuen String "*filename*" geschrieben werden. Dieser wird anschließend durch die "*Split*('_')" Anweisung unterteilt. Dadurch kann der übergebene Präfix mit dem "MainPräfix" im String-Array "*filenameFront*[0]" verglichen werden. Auch muss überprüft werden, ob im zweiten Teil, dem "SubPräfix", eine "*low−*" vorhanden ist. Das ist deshalb wichtig, da alle Bilder zweimal im Cloud-Speicher liegen. Einmal in hoher und einmal in niedriger Auflösung. Für eine Vorschau der Bilder werden allerdings nicht beide benötigt.
Sind beide Vergleiche positiv, wird der Name der Datei mit dem Schlüssel und dem Containernamen mit einem "_" unterteilt in die "*strListe*" geschrieben. Diese wird an das Model übergeben und die "*FindImage*" -View aufgerufen.

```
public ActionResult FindImage(string container, string praefix, string key)
{
    List<string> strListe = new List<string>();
    if ((container == null) || (praefix == null) || (key == null))
    {
      return View(strListe);
    }

    CloudBlobContainer blobContainer =
        _blobStorageService.GetCloudBlobContainer(container);
    List<string> blobs = new List<string>();

    foreach (var blobItem in blobContainer.ListBlobs())
    {
      blobs.Add(blobItem.Uri.ToString());
    }


    if (blobs.Count < 1)
    {
      blobContainer.Delete();
      return View(strListe);
    }

    foreach (var item in blobs)
    {
      string filename = item.Substring(Math.Max(0, item.Length - 18));
      var filenameFront = filename.Split('_');
      if ((filenameFront[0].Equals(praefix))&& filenameFront[1].Contains("low-"))
      {
        strListe.Add(filename + "_" + key + "_" + container);
      }
    }
    return View(strListe);
}
```

Listing 5.10: Verweisen der Bilder - FindImage()

Die "*FindImage*" -View liest die Daten aus dem Model in eine String-Liste "*urlList*" ein. Sind in dieser Liste keine Einträge vorhanden, wird dem Benutzer eine Meldung angezeigt,

dass Keine Bilder gefunden wurden, da diese entweder bereits gelöscht wurden, oder der Link fehlerhaft ist. Siehe Listing 5.11

Bei vorhandenen Einträgen, wird für jeden Eintrag durch ein Image-Tag, über das Source-Attribut die Aktion-Methode "*BildLow*" des Home-Controllers abgerufen, welche die Größe des Bildes festlegt. Dem Aufruf über "*@Url.Action*()" wird hierzu der String-Eintrag aus der "*urlList*" mit übergeben. Anhand dieser Parameter benutzt die Aktion-Methode "*BildLow*" die "*LoadPic*" -Funktion zum Laden des Bildes und die "*AES_Decrypt*" -Funktion zum Entschlüsseln und liefert das Bild an die View zurück. Dieses wird dadurch als Vorschau in kleinem Format dargestellt, durch die Definition von "*height* =″ 200″" und "*width* =″ 200″".

Durch das "< *a href* =″ ″ >" -Attribut wird zu jedem Bild in Form eines Linkes des Herunterladen in hoher und niedriger Auflösung angeboten. Diese Aufrufe für das Herunterladen erfolgt wie der zur Erstellung des Bildes zur Vorschau mithilfe des Image-Tags. Zudem wird für jedes Bild ein "*@Hml.ActionLink*()" generiert. Dieser ruft die "*Delete*()" -Funktion auf und übergibt ebenfalls den String-Eintrag aus der "*urlList*". Der Aufruf enthält darüber hinaus einen weiteren booleschen Parameter, der für die Entfernung der einzelnen Bilder auf "*false*" gesetzt ist. Dieser Parameter dient für die "*Delete*()" zur Erkennung, ob nur ein einzelnes Bild gelöscht werden soll, oder alle Bilder des aktuellen Präfixes.

Unterhalb aller gelisteten Bilder befindet sich eben dieser Action-Link, der es ermöglicht, alle Bilder auf einmal zu entfernen, wie auf der Abbildung 5.2 ersichtlich.

```
var urlList = (List<string>)Model;
if (!urlList.Any())
{
        <p>Es wurden keine Bilder gefunden.</p>
        <p>Entweder wurde das Bild bereits entfernt, oder Ihr Link ist
            fehlerhaft.</p>
}
else
{
   foreach (var url in urlList)
   {
     <table><tr><th>
       <img src="@Url.Action("BildLow", "Home", new {strAll = url})" height="200"
           width="200"/>
     </th><th></th><th>
       <a href="@Url.Action("BildLow", "Home", new { strAll = url })"
           download="Bild.jpg"> Download (niedrige Auflösung)</a><br/>
       <a href="@Url.Action("BildHigh", "Home", new { strAll = url })"
           download="Bild.jpg"> Download (hohe Aufösung)</a><br/>

       @Html.ActionLink("Bild löschen...", "Delete", new { strAll = url , flag =
           false })
     </th></tr></table>
   }

   @Html.ActionLink("Alle Bilder löschen...", "Delete", new { strAll =
       urlList.First(), flag=true})
```

```
}
```

Listing 5.11: Verweisen der Bilder - FindImageView()

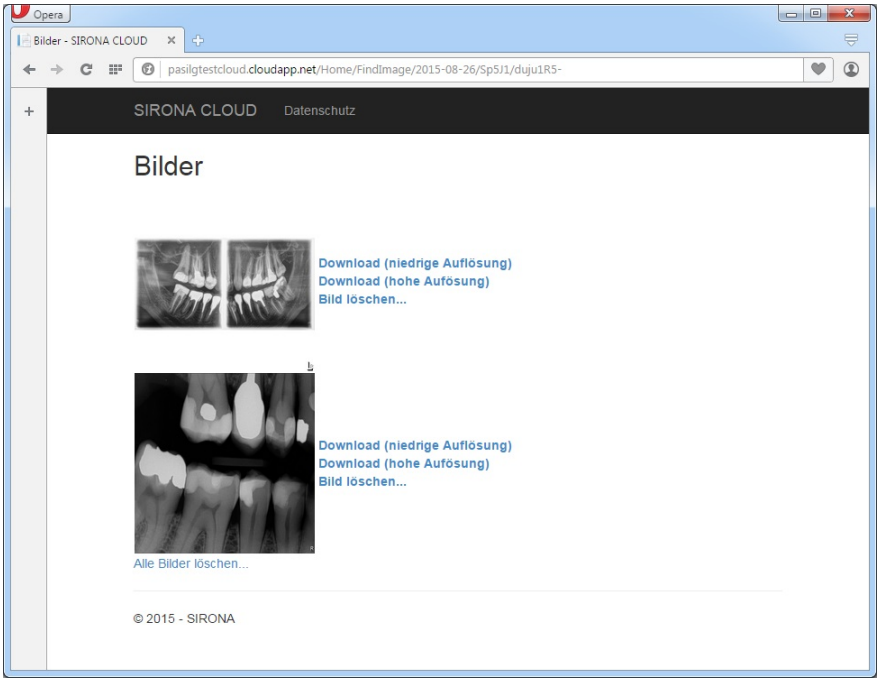


Abbildung 5.2: Verweisen der Bilder - Webseiten Darstellung

5.2.2 Automatische Bildbereinigung

Für die Automatische Bildbereinigung der Bilder auf dem Azure-Speicher, wurde ein Programm "CleanUp" programmiert. Da die Container auf dem Azure-Speicher alle im Format des Datums angelegt sind, beruht die Logik der Bildentfernung darin, das aktuelle Datum auf 7 Tage zurück zu rechnen und mit dem Namen des Containers zu vergleichen. Um mit dem Datum rechnen und vergleichen zu können wird der Typ DateTime verwenden.

Mithilfe der "BlobStorageService" -Klasse werden alle existierenden Containernamen in eine Liste gefüllt. Jeder Container-Name wird anschließend versucht über die "DateTime.TryParseExact()" -Methode in ein gültiges Datum in dem Format "$yyyy - MM - dd$" umzuwandeln.

Um zu überprüfen, ob der Container bereits älter als 7 Tage ist, stellt "DateTime" eine Funktion "$DateTime.Compare()$" zur Verfügung. Diese vergleicht zwei Instanzen vom Typ "DateTime" und liefert die Werte kleiner als 0, 0 und größer als 0 zurück. Liegt der Name als Datum des Containers vor dem zu vergleichenden Datum, wird der Container über den Namen referenziert und mit samt dem Inhalt gelöscht, wie in Listing 5.12 dargestellt.

```
static void Main(string[] args)
{
    BlobStorageServices blobStorageService = new BlobStorageServices();
    string folder = DateTime.UtcNow.ToString("yyyy-MM-dd");
    var date = DateTime.ParseExact(folder, "yyyy-MM-dd",
        CultureInfo.InvariantCulture);
    date = date.AddDays(-7);
    var containers = blobStorageService.GetCloudBlobContainers();

    foreach (var item in containers)
    {
          DateTime name;
          if (DateTime.TryParseExact(item.Name, "yyyy-MM-dd",
              CultureInfo.InvariantCulture, DateTimeStyles.None, out name))
          {
             var result = DateTime.Compare(name,date);
             if (result < 0)
             {
                  var container =
                      blobStorageService.GetCloudBlobContainer(item.Name);
                 container.Delete();
             }
          }
    }
}
```

Listing 5.12: Automatische Bildbereinigung - CleanUp Code

Über das Portal der Microsoft-Azure-Cloud, wird das CleanUp Visual-Studio Projekt im ZIP-Dateiformat hochgeladen und konfiguriert. Dazu wird ein neuer Web-Auftrag angelegt. Ein Web-Auftrag führt serverseitig Anwendungen automatisch aus und kann bezüglich dem Zeitplan unterschiedlich konfiguriert werden, sie Abbildung 5.3.

Der Zeitplan erfolgt periodisch mit einem Wiederholungsintervall von einem Tag. Gestartet wird dieser jeden Abend um 23:30 Uhr, nach Coordinated Universal Time (UTC) + 02:00. Dieser kann auch jederzeit manuell über das Azure-Portal gestartet werden.

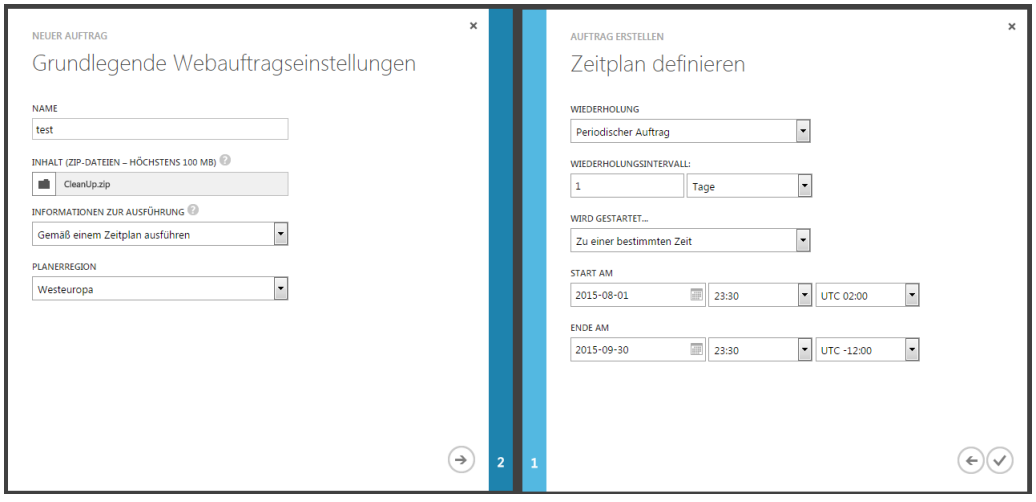


Abbildung 5.3: Automatische Bildbereinigung - Konfiguration

6 Fazit

Das Kapitel Fazit beinhaltet eine Zusammenfassung der Thesis, ein persönliches Fazit, sowie einem Ausblick mit Möglichkeiten zur Verbesserung und Weiterentwicklung des implementierten Prototypen.

6.1 Zusammenfassung

In dieser Arbeit wurde ein Vorgehen zur Entwicklung eines Prototypen für die Bereitstellung von Bilddaten für die Patientenkommunikation evaluiert und entwickelt. Der Prototyp wurde als Modul in der bereits vorhandenen Software SIDEXIS 4 implementiert.

Dazu wurden die rechtlichen Vorgaben des Datenschutzes zum Umgang und Auslagern von Patientendaten genauer untersucht. Es wurde festgestellt, dass unter gewissen Umständen durch Verschlüsselung und Pseudonymisierung der Bilddaten, sowie der Einwilligung des Patienten die Auslagerung durchaus gesetzeskonform durchgeführt werden kann. Daraus ließen sich Anforderungen festlegen die sowohl für den Datenschutz, als auch teilweise für die Technologie elementar sind. Hinzu kamen weitere funktionelle Vorgaben und Anforderungen, welche die Benutzerinteraktion und auch die Performance und Bildbereitstellung betreffen.

Anhand der Anforderungen wurden die Anwendungsansätze festgelegt. Diese beinhalten Grundfaktoren zur Berechnung der Kosten der Cloud-Systeme, sowie dem Speicherplatzbedarf. Die Grundfaktoren ließen sich anhand Statistiken und Schätzungen diverser Produktmanager definieren. Außerdem wurde gemäß den Anforderungen drei Cloud-Systeme, Azure, Amazon und Strato zur näheren Evaluierung festgelegt.

Bei der Evaluierung der Cloudsysteme im einzelnen, wurde zu den Bereichen der Technologie, dem Datenschutz, der Zukunftssicherheit und den Kosten, detailliertere Informationen zusammengetragen. Auf der Grundlage dieser Informationen konnten in der Gesamtbetrachtung die Vor- und Nachteile im einzelnen gegenübergestellt und bewertet werden. Diese wurden in Form einer Matrix dargestellt, welche zur Entwurfsentscheidung beitrugen.

Daraufhin erfolgte in der Entwurfsentscheidung eine Festlegung, mit welchen Systemen und in welcher Form der Prototyp umgesetzt werden soll. Hier wurde aufgrund der Gesamtbetrachtung und der Evaluierung zuvor, auf das Cloud-System von Azure als Online-Plattform

gesetzt. Die Umsetzung der Implementierungen sollte aufgrund der bereits bestehenden Software SIDEXIS 4 mit C# in Visual-Studio verrichtet werden.

Nachdem die Entscheidungen für den Entwurf festgelegt wurden, konnte ein Ablauf der Funktionsweise des Prototypen entworfen werden, sowie eine Übersicht des Systems und eine Abgrenzung der benötigten Klassen und Komponenten. Eine Beschreibung dazu wurde in dem Kapitel Beschreibung des Systems erstellt. Darin wurde im Design des Moduls in SIDEXIS 4 und der Azure Cloud unterschieden. Zudem wurde die Schnittstelle, welche die beiden Komponenten verbindet, genauer beschrieben.

Zuletzt wurde in dem Kapitel der Implementierung die genauere Umsetzung der Programmierung einiger bestimmter signifikanter Vorgänge beschrieben. Diese betrafen seitens des SIDEXIS 4 Moduls die AES-Verschlüsselung, sowie das hochladen der Bilder. Für die Programmierung der Azure-Cloud wurden der Verweis der Bilder und die automatische Bildbereinigung mit deren Konfiguration detaillierter erläutert. Der Verweis der Bilder ist vor allem aufgrund der Schnittstellenverbindung ein Dreh- und Angelpunkt der Funktionalität.

6.2 Fazit

Zusammenfassend ist ein Plattformunabhängiger Austausch von Bilddaten für die Patientenkommunikation in Anbetracht des Datenschutzes möglich. Dass viele Unternehmen die allgemeine Auslagerung von Daten in eine Cloud eher kritisch betrachten ist verständlich. Allerdings kann mit den richtigen Sicherheitsaspekten, wie der Verschlüsselung und zertifizierten Verbindungen, unter der richtigen Anwendung die Sicherheit der Daten garantiert werden. Durch die Verschlüsselung der Daten, kann ein Angreifer im Falle einer Kompromittierung der Cloud nichts mit den Bilddaten anfangen, da dieser nicht in Besitz der Schlüssel zum entschlüsseln gelangen kann.

Da ein Zertifikat der Firma Sirona zu dem Zeitpunkt der Implementierung zwar vorhanden war, allerdings noch nicht zur Verfügung stand, ist die Funktionalität des Prototypen auf die Standard HTTP-Verbindung beschränkt. Dies gilt nur für das Herunterladen der Bilder. Das Hochladen in den Azure-Speicher erfolgt über HTTPS.

Das überaus große Angebot von Cloud-Anbietern zeigt meiner Meinung nach, dass das Interesse an der Nutzung zur Auslagerung von Daten in die Cloud vorhanden ist und immer mehr genutzt wird. Nicht zuletzt ist aus wirtschaftlicher Sicht das Auslagern der IT-Infrastruktur auf längere Sicht günstiger, als eigene Systeme bereitstellen und warten zu müssen. Die Entscheidung für eines der Cloud-Systeme war etwas anspruchsvoller als erwartet. Da die meisten Anbieter kaum Unterschiede aufweisen, gestaltete sich die Recherche nach wichtigen

Differenzen mühseliger als gedacht.

Als Fazit dieser Thesis ist festzuhalten, dass die Digitalisierung und Auslagerung der Bilddaten in der heutigen Zeit, mit dem Smartphone als Trend ein innovativer Fortschritt ist, der Patienten eine moderne und angenehme Bereitstellung seiner Röntgenbilder ermöglicht. Die Einfachheit durch welche der Patient seine Bilder erhält ist meiner Meinung nach komfortabler und flexibler als die Varianten des Drucks oder dem Brennen auf CDs und ist ansprechend für die aktuellen Generationen.

6.3 Ausblick

Die Umsetzung des Prototypen hat gezeigt, dass eine Online-Plattform zum Austausch von Patientendaten durchaus möglich ist und auf verschiedene Weisen implementiert werden kann.
Implementierungen die dieser Prototyp noch nicht erfüllt, sind eine sichere HTTPS-Verbindung zwischen dem Benutzer und der Azure-Cloud-Webseite, um einen sicheren Herunterladen der Bilddaten garantieren zu können. Diese kann mittels einem Zertifikat umgesetzt werden.

Eine Weitere Möglichkeit zur Verbesserung der Sicherheit wäre die Entschlüsselung der Bilder im Browser. Da in den Anforderungen festgelegt wurde, dass keine Software zur Entschlüsselung auf Seitens des Benutzers benötigt werden soll, wurde die Entschlüsselung serverseitig vorgenommen um die Bilddaten anschließend an den Benutzer zu übertragen. Dadurch hat der Cloud-Betreiber kurzzeitig Zugriff auf die entschlüsselten Bilddaten, wenn auch nur im flüchtigen Speicher. Mit einer Entschlüsselung im Browser, wäre das nicht der Fall und der Benutzer benötigte auch keine zusätzliche Software zur Entschlüsselung.

Auch das Design der Webseite sowie die des Moduls können in ihrer Anschaulichkeit noch verbessert werden. Ressourcen für die unterschiedlichen Sprachen sind auf der Cloud-Webseite noch nicht vorhanden und müssten noch in die Ressourcen übersetzt und eingepflegt werden.

Abbildung 6.1: QR-Code Verweis - gültig bis 2015, Sep. 15

Anhang

A.1 Pflichtenheft

D 0001

Plattformunabhängiger Austausch von Bilddaten für die Patientenkommunikation

[redacted] **GX**

Pflichtenheft

(*Design – Input*)

Version 0.1

Status:

	Erstellung / ***Issuance:***	**Prüfung** / ***Verification:***	**Freigabe** / ***Approval:***
Name / *Abtlg. /* Name / *Division:*	***Pascal Ilg***	Nicht erforderlich / *Not required*	
Datum / *Date:*			
Unterschrift / *Signiture:*			

Historie / *History*

Ver-sion	**Datum** *Date*	**Ersteller / Abt.** *Provided / Dep.*	**Grund der Änderungen** *Changes and motivation*
0.1	19.06.2015	Pascal Ilg / GBE	Neuerstellung

Inhaltsverzeichnis

Tabelle der Anforderungsschlüssel

1 Einleitung

1.1 Zweck des Dokuments

Dieses Pflichtenheft beschreibt die bestehenden Anforderungen der betroffenen Zielgruppen für den Austausch von Bilddaten, auf einer unabhängigen Plattform zur Patientenkommunikation.

Schwerpunkt liegt dabei auf der Sicherheit, derTechnologie und dem Datenschutz, sowie der komfortable Nutzung für den Kunden.

1.2 Definitionen, Akronyme und Abkürzungen

1.2.1 Definitionen

1.2.2 Akronyme und Abkürzungen

FA	Funktionale Anforderungen
HA	Hygieneanforderungen
KA	Kundenanforderungen
QA	Qualitätsanforderungen
UA	Umweltanforderungen
WPP	Wirtschaftlicher Produktplan
ZA	Zulassungsanforderungen

1.3 Referenzdokumente

(nicht zutreffend)

2 Allgemeine Beschreibung

2.1 Generelle Projektbeschreibung

Derzeit werden Bilddaten in Zahnarztpraxen dem Patienten bei Anfrage in Form einer Kopie oder digital auf einer CD zur Verfügung gestellt, für welche auch teilweise Kosten anfallen können. Mittlerweile gibt es allerdings kostengünstigere und bequemere Wege für den Patienten.
Ziel ist es eine Erweiterung in die bestehende Software SIDEXIS 4 zu implementieren, welche die Möglichkeit bietet, Röntgenbilder, verschlüsselt mit einer gesicherten Verbindung, hochzuladen und den Link zum Bild in Form eines QR-Codes darzustellen. Der Patient kann die Bilddaten durch Abfotografieren des QR-Codes mit seinem Smartphone und einer QR-Code-Reader App herunterladen. Der Patient soll zudem zwischen zwei Unterschiedlichen Auflösungen der Bilddaten entscheiden, welche er herunterladen möchte. Als Alternative kann der Hyperlink auch als Email an den Kunden geschickt werden.
Die Bilddaten auf der Plattform sollen nur eine bestimmte Zeit lang bereitgestellt werden nachdem sie dann wieder automatisch entfernt werden. Der Patient hat allerdings die Möglichkeit auf Wunsch die Bilddaten schon vorher jederzeit vom Server zu entfernen.

Abgeleitet von dem Szenario, dass Röntgenbild vom Monitor abzufotografieren, soll dieser Ablauf ebenso bequem, und ohne Hemnisse für den Patienten zur Bereitstellung von hochwertigen Bilddateien erfolgen.

2.2 Schlüssel – Kundenanforderungen

2.3 Zielgruppe

Zielgruppe sind alle Patienten, die von Arztpraxen mit der SIDEXIS 4 Software behandelt werden und Röntgenbilder erzeugen.

Ärzte

Für Ärzte in der Zahnmedizin bietet diese Patientenkommunikation eine erhebliche Erleichterung durch automatisiertes Hochladen der Bilddaten und auch wieder entfernen. Dabei verbindet diese Erweiterung keine Einschränkungen oder Kosten für die Praxis.

Patient

Alle Patienten im Besitz eines Smartphones und einer „QR-Reader App“ mit Internetzugang, bekommen die Möglichkeit, ihre Bilddaten digital herunter zu laden und weiter zu verarbeiten. Außerdem kann sich jeder Patient mit einer gültigen E-Mail Adresse, den Link zum Herunterladen zuschicken lassen.

2.4 Markpositionierung

(nicht zutreffend)

2.5 Lebenszyklus

Die geschätzte Lebensdauer der Erweiterung ist abhängig von dem Lebenszyklus von SIDEXIS 4

2.6 Projektkostenrahmen

(nicht zutreffend)

2.7 Prioritäten

Das Projekt hat folgende Projektprioritäten:

1. Sicherheit der Bilddaten (Datenschutz)
2. Technlogie
3. Bequeme Patientenkommunikation

2.8 EVA

(nicht zutreffend)

3 Beschreibunge des Produktes

3.1 Funktionelle Anforderungen

Hemmnisse für den Patienten

Für den Patienten sollen keine Hemnisse zum Erlangen der Bilddaten entstehen. Dies beinhaltet keine Registrierung oder Passworteingabe beim Herunterladen. Außerdem sollen keine weiteren Kosten für den Patienten entstehen.

Internetzugang

Der Patient benötigt einen Internetzugang über mobile Daten oder einen WLAN-Zugang.

Personenbezogene Daten

Es soll kein Zugriff auf personenbezogenen Daten erfolgen oder Code implementiert werden welcher dem Patienten schaden könnte durch abfotografieren des QR-Codes.

Alternative

Alle Patienten, die nicht der entpsrechenden Zielgruppe zutreffen, sollen die Möglichkeit haben, einen Link zum Download der Bilddaten per Email zu bekommen, sofern eine Emailadresse vorhanden ist.

Verfügbarkeit

Die Verfügbarkeit der Systeme sowie der Einsatz sollen langfristig gewährleistet sein.

Bildqualität

Die Bilddaten werden in einer minimalen Auflösung von 1024 x 768, mit einer Größe von 100 KiloByte bis 250 KiloByte und in der Originalauflösung von ca. 1920 x 1080 mit einer maximalen Größe von 1 MegaByte angeboten werden.

Plattformunabhängig

Die Bereitstellung der Bilddaten soll Plattformunabhängig im JPEG Format erfolgen. Der QR-Code soll ebenfalls von jedem standartmäßigem QR-Code Reader gelesen werden können.

Integrität

Die Bilddaten sollen mit AES 256 Bit verschlüsselt werden. Jeder Patient bekommt für sein Bilddaten einen eigenen Schlüssel zum entschlüsseln. Es soll lediglich nur der Person möglich sein, die Bilddaten einzusehen, welche den entsprechenden Link oder QR-Code besitzt.

Vertraulichkeit

Die Verbindung beim Hochladen, sowie beim Herunterladen soll durch Hypertext Transfer Protocol Secure HTTPS) gesichert sein.

Bereitstellung

Wenn es der Patient erlaubt, sollen die Bilddaten für bis zu 7 Tage in der Cloud bereitgestellt werden, nachdem diese hochgeladen wurden. Nach Ablauf dieser Zeit werden diese wieder entfernt. Es soll im Nachhinein möglich sein, den Zeitraum über die Verfügbarkeit der Bilddaten auf der Cloud anzupassen.

Entfernen

Die Bereinigung der Bilddaten soll einmal täglich automatisiert, Server-seitig ablaufen. Der Patient hat die Möglichkeit, die Bilddaten bereits jederzeit früher zu entfernen.

Datenschutz

Die Rechte der hochgeladenen Bilddaten sollen nicht an den Anbieter der Plattform abgetreten werden.
Der Patient kann die Datenschutzerklärung einsehen und wird vor dem Abfotografieren des QR-Codes durch einen Disclaimer darauf hingewiesen.

Lokalisierung

Der Patient bekommt die Benutzeroberfläche der HTML-Seite in eine der folgenden Sprachen angezeigt:

- Deutsch
- Englisch

Es ist nicht spezifiziert ob weitere Sprachen nachträglich ohne Code-Änderungen ergänzt werden können.

3.2 Klinische Anforderungen

(nicht zutreffend)

3.3 Hygiene

(nicht zutreffend)

3.4 Design

Das Design der Software SIDEXIS 4 wird verwendet.

3.5 Ergonomie

(nicht zutreffend)

3.5.1 Spezifikation der Anwendung des Moduls

Die Erweiterung an SIDEXIS 4 ist dazu da, Bilddaten zur Patientenkommunikation Online bereitzustellen, wobei der Datenschutz gewahrt bleiben muss.

3.5.2 Hauptbedienfunktionen des Moduls

- Bereitstellen von Bilddaten auf einer Online-Plattform
- Generieren einer Verlinkung zu den Bilddaten, in Form eines QR-Codes oder per Email.

3.5.3 Häufig benutzte Funktionen

(nicht zutreffend)

3.5.4 Spezifikation der Gebrauchstauglichkeit

(nicht zutreffend)

3.6 Strahlung

(nicht zutreffend)

3.7 Q-Daten

(nicht zutreffend)

3.8 Wartung und Service

Das System soll Wartungsfrei laufen. Gegebenfalls können die Kosten der Plattform durch kürzere bereitstellung angepasst werden.

3.9 Garantie

Keine weitergehende Garantierleistungen.

3.10 After Sales Service

(nicht zutreffend)

3.11 Betriebs- Lager- und Transportbedingungen

(nicht zutreffend)

3.12 Zulassung und Normen

(nicht zutreffend)

3.13 Prüf- / Fertigungskonzept

(nicht zutreffend)

3.14 Installations- / Schnittstellenbedingungen

3.14.1 Benutzerschnittstellen

3.14.1.1 Performance

Upload-Performance

Bei einem Internetzugang mit einer Upload-Geschwindigkeit von 576 kbit/s und Bilddaten mit der Größe von insgesamt 750 kb, beträgt die Wartezeit maximal 10 Sekunden.

QR-Code-Performance

Die Erstellung des Qr-Codes soll unter 10 Millisekunden betragen.

Download-Performance

Bei einem Internetzugang von einer Download-Geschwindigkeit von 6016 kbit/s, soll die Wartezeit der Bilddaten, mit der Größe von insgesamt 750 kb, 997 Millisekunden betragen.

Lösch-Performance

Das Löschen der Bilddaten hängt von der Belastung des Webservers ab. Im Durchschnitt benötigt ein Server mit 1 CPU und 1 GB RAM für die Durchführung einer Anfrage < 0,1 Millisekunden.

3.15 Umweltbedingungen

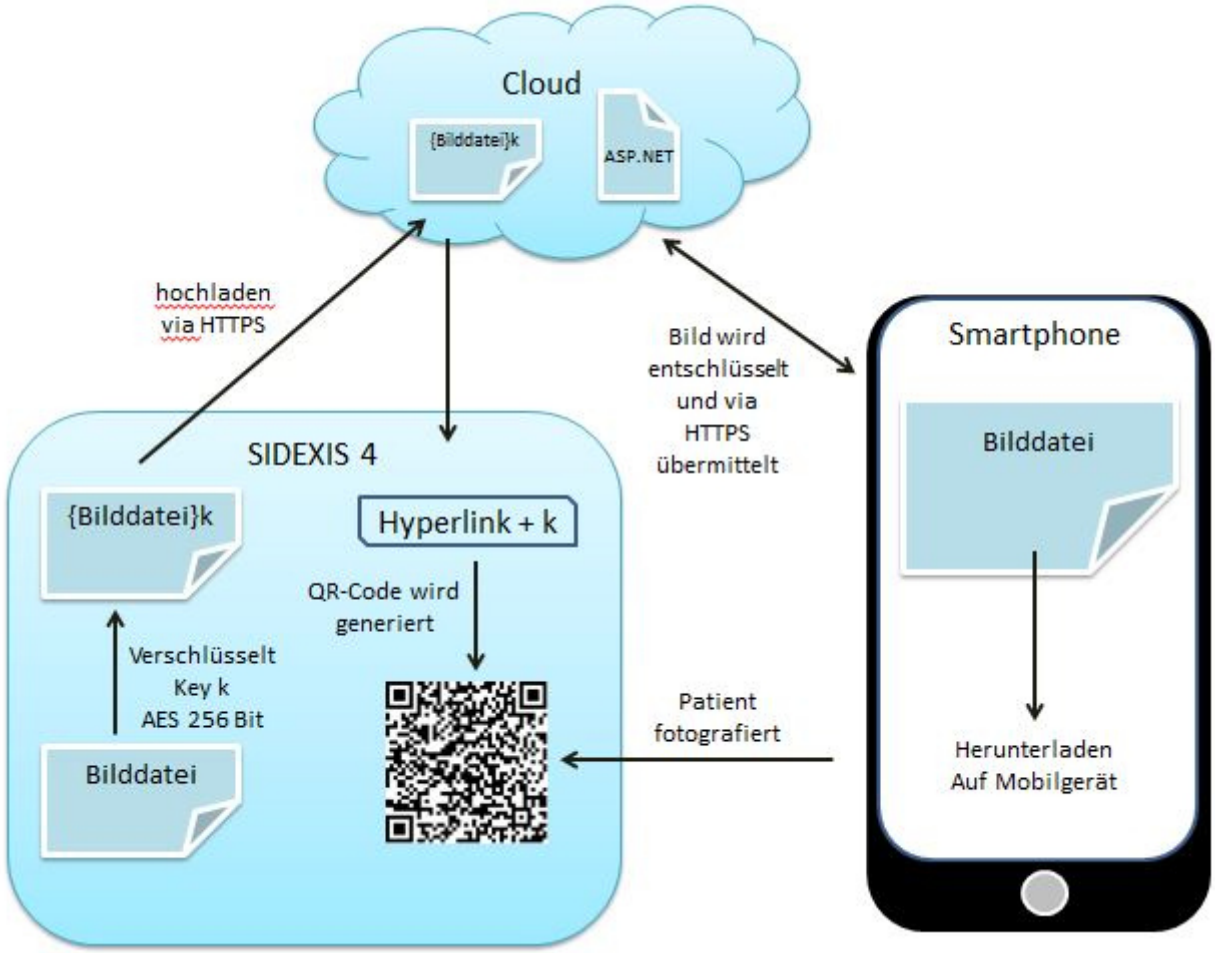
(nicht zutreffend)

4 Beschreibung des Systems

Das System besteht aus mehreren Komponenten. Die Software SIDEXIS 4, dem OnlineService bestehend aus einer Cloud und einem Webserver, und einem Smartphone.

SIDEXIS 4 verschlüsselt zunächst die Bilder und lädt sie über eine HTTPS-Verbindung in die Cloud. Dazu wird ein Hyperlink generiert mit dem Schlüssel „k“ als Übergabeparameter. SIDEXIS 4 generiert zu diesem Hyperlink einen QR-Code. Dieser kann mithilfe eines QR-Code Reader abfotografiert werden.

Der Link führt dann zu dem Webserver, welcher die Daten von der Cloud abgreift, sie entschlüsselt und dem Client als Download zur Verfügung stellt.



5 Beschreibung u. Spezifikation an Baugruppen

(nicht zutreffend)

6 Anforderungsspezifikationen für Software

(nicht zutreffend)

7 Beschreibung der Benutzerinteraktionen

7.1 Eula/Checkbox

In SIDEXIS 4 wird zunächst nur der Workspace mit den Bildern angezeigt. Der Patient Akzeptiert durch Checken einer Eula-Checkbox und Bestätigen der Datenschutzerklärung durch einen Button den Vorgang zum Hochladen der Bilddaten in die Azure Cloud.

7.2 Abfotografieren

Der Benutzer kann mit seinem Smartphone und einer QR-Reader-App den QR-Code abscannen und wir auf eine Webseite verwiesen, auf der seine Bilddaten in den verschiedenen Bildformaten zum Download angeboten werden.

A.2 Berechnungen der Kosten

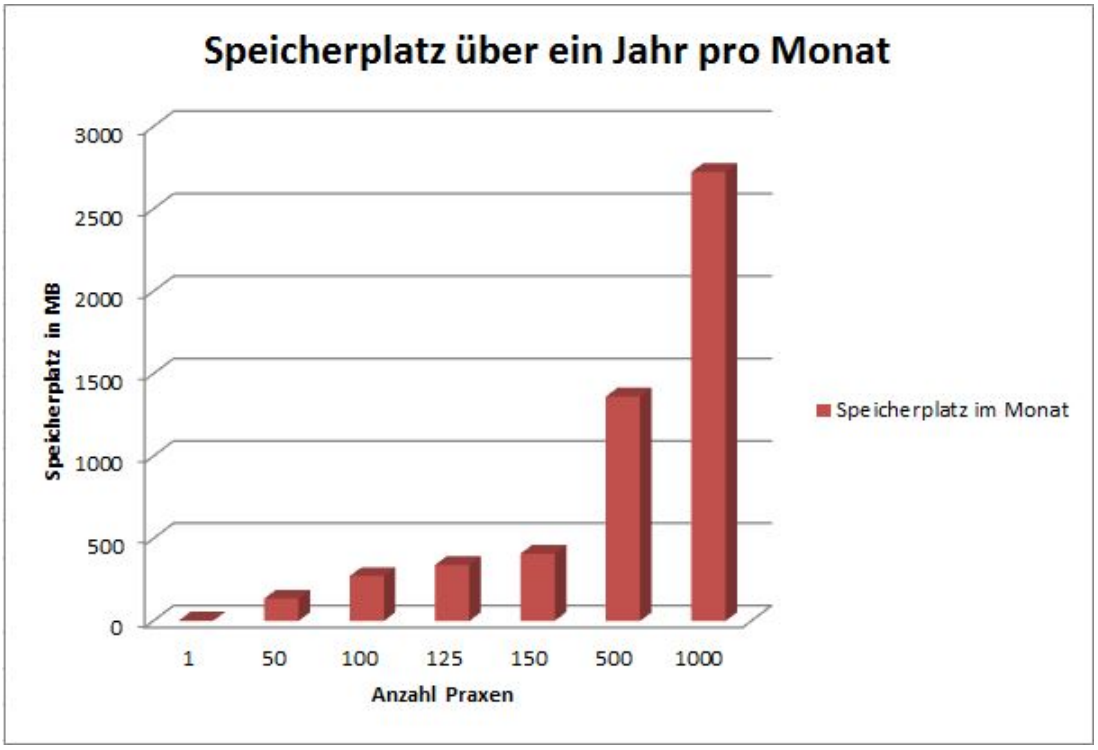


Abbildung A.1: Ergebnis der Berechnung des Speicherplatzbedarfs im Monat

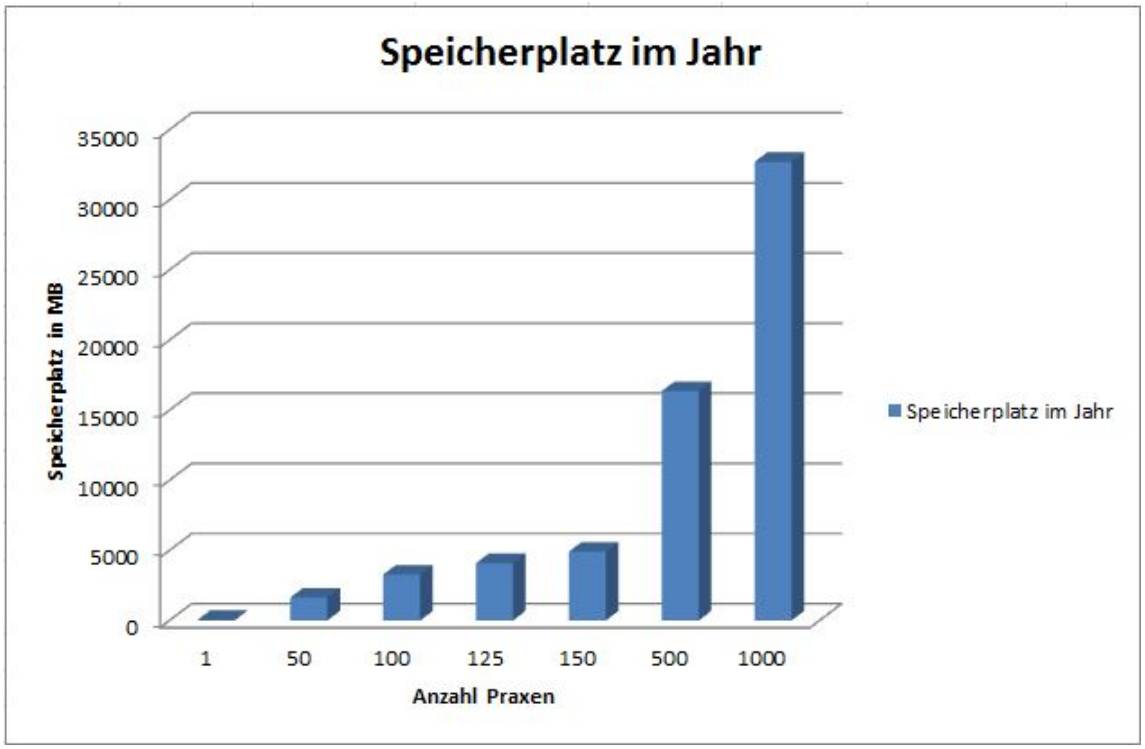


Abbildung A.2: Ergebnis der Berechnung des Speicherplatzbedarfs im Jahr

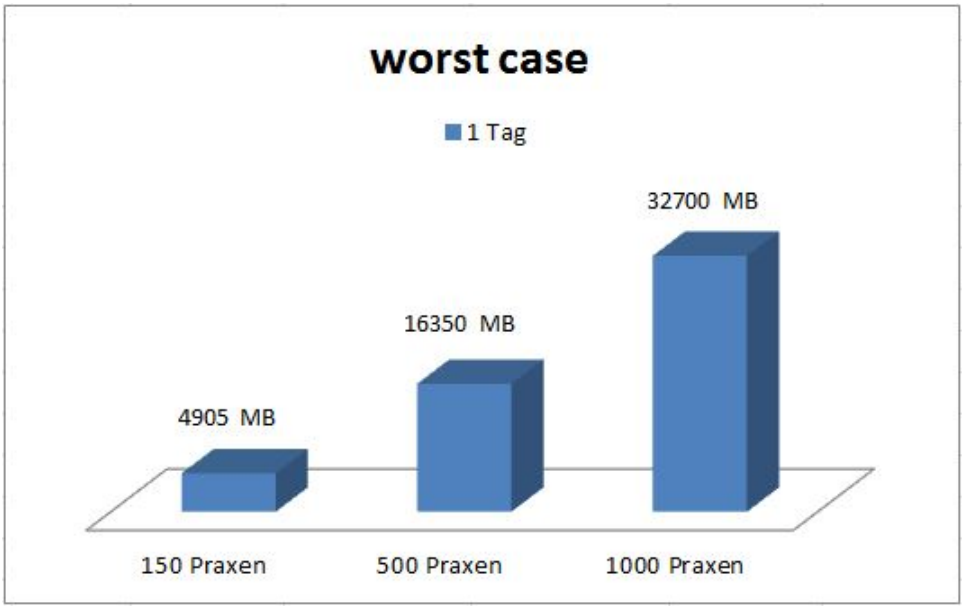


Abbildung A.3: Ergebnis der Worst-Case Berechnung des Speicherplatzbedarfs

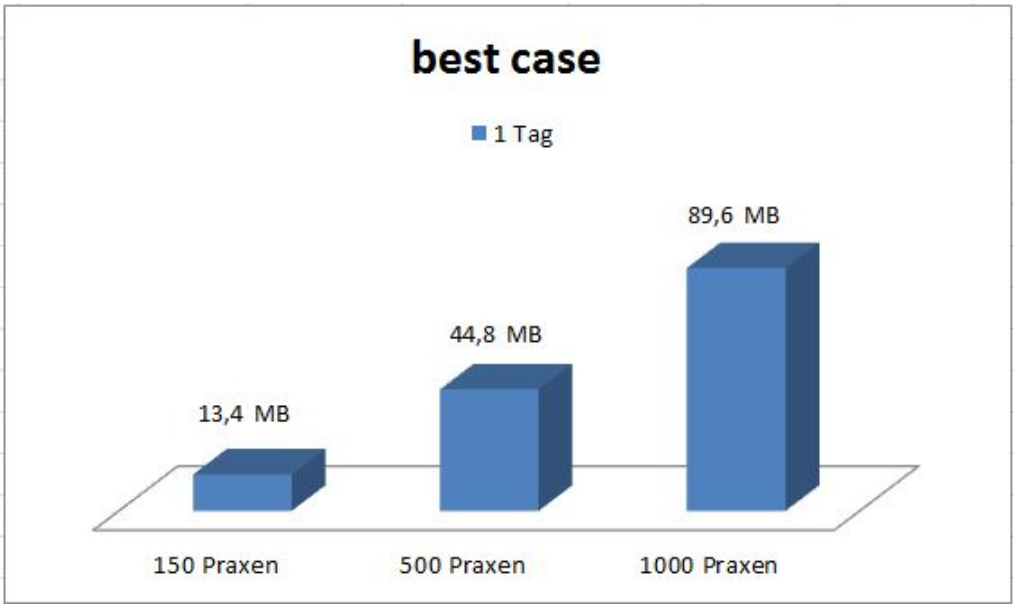


Abbildung A.4: Ergebnis der Best-Case Berechnung des Speicherplatzbedarfs

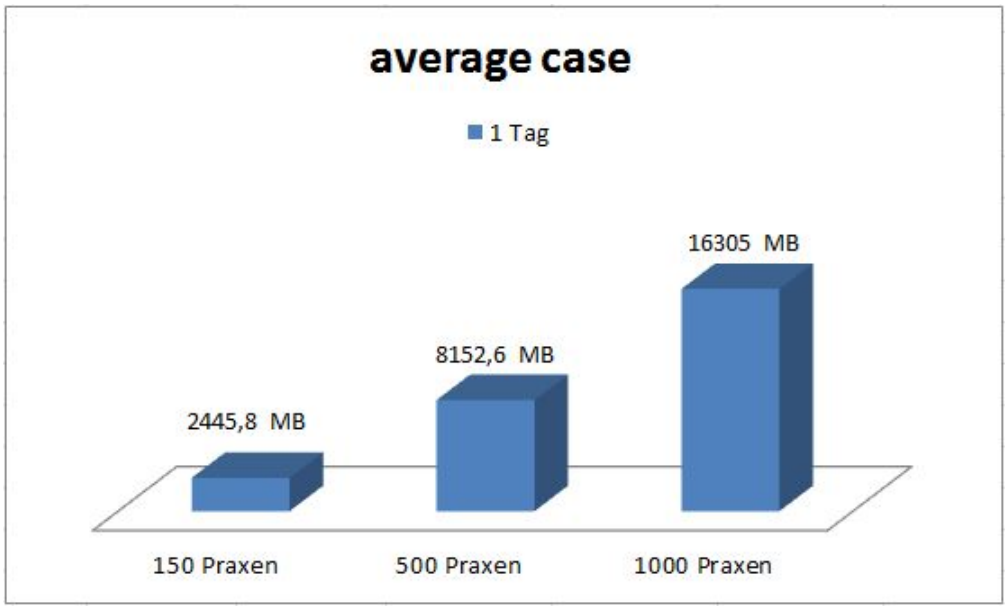


Abbildung A.5: Ergebnis der Average-Case Berechnung des Speicherplatzbedarfs

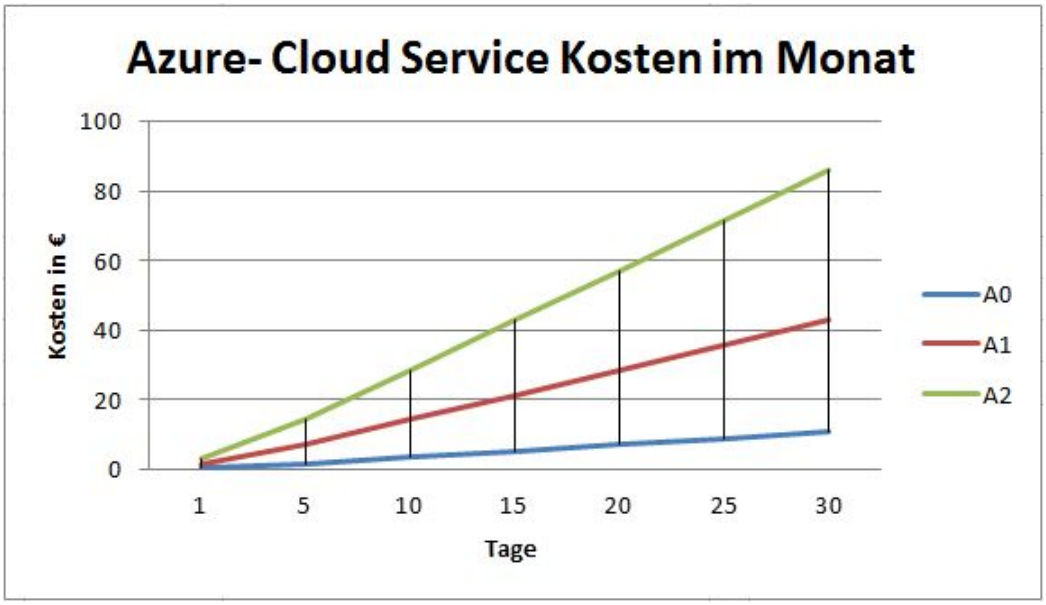


Abbildung A.6: Ergebnis der Azure-Cloud-Service Kosten im Monat

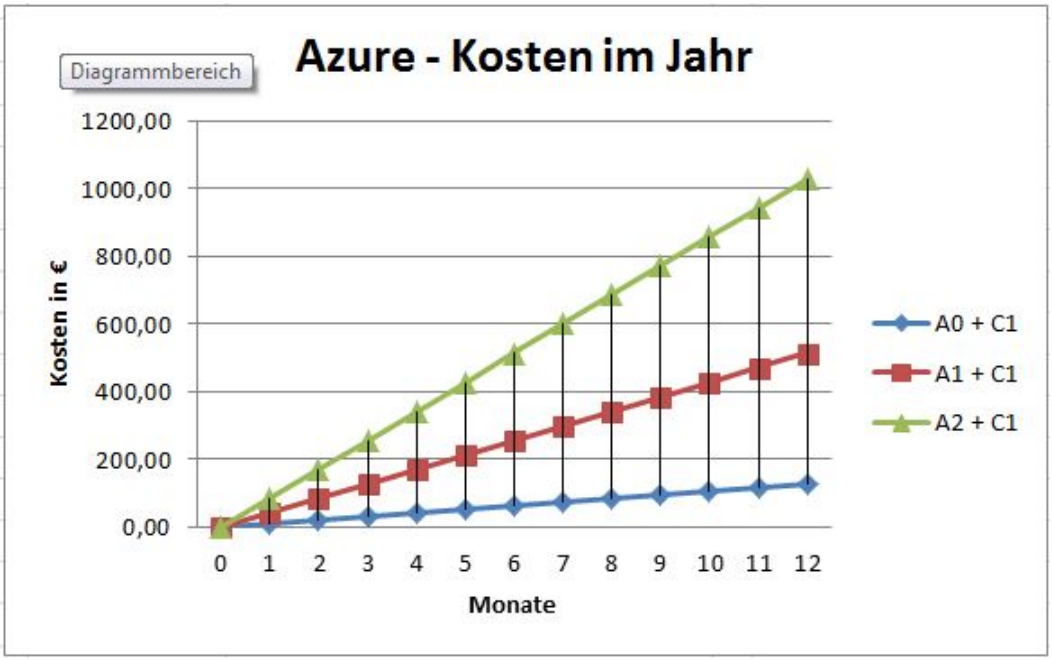


Abbildung A.7: Ergebnis der Azure Kosten im Jahr

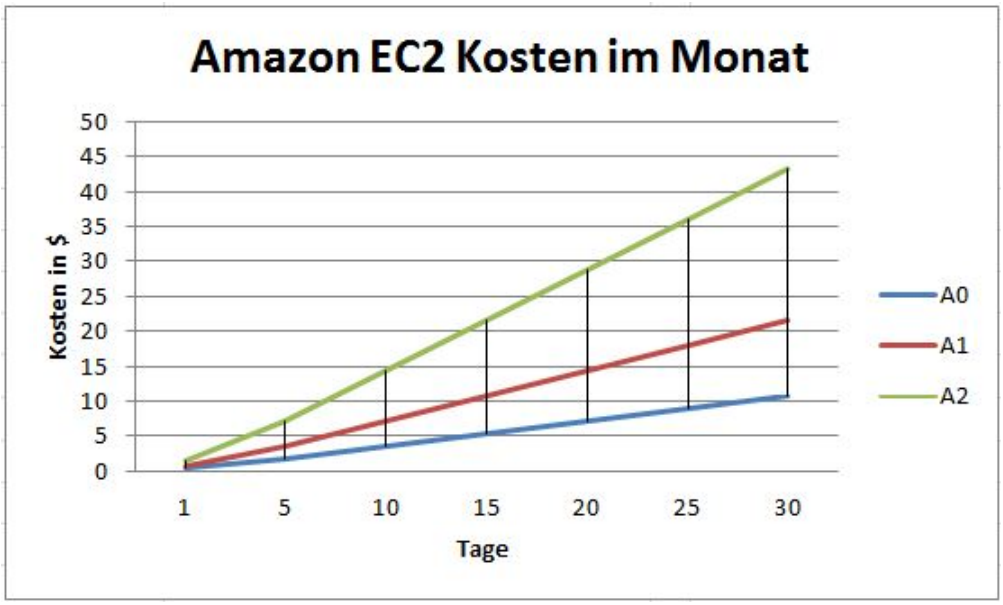


Abbildung A.8: Ergebnis der Amazon EC2 Kosten im Monat

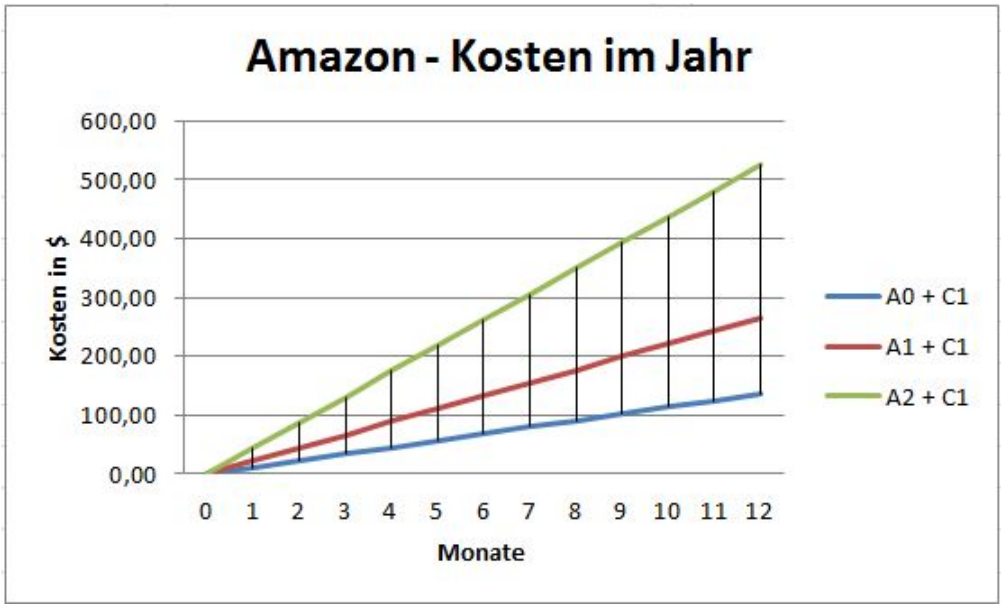


Abbildung A.9: Ergebnis der Amazon Kosten im Jahr

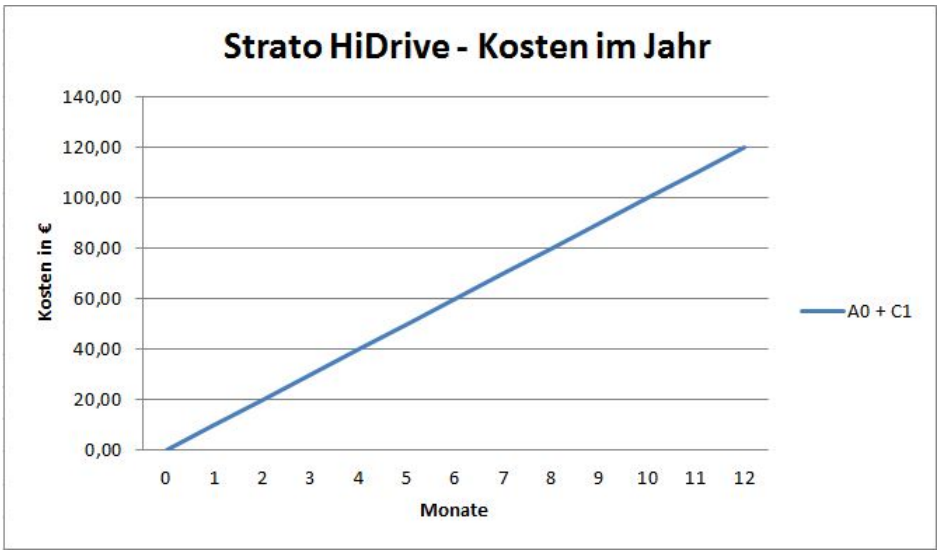


Abbildung A.10: Ergebnis der Strato Kosten im Jahr

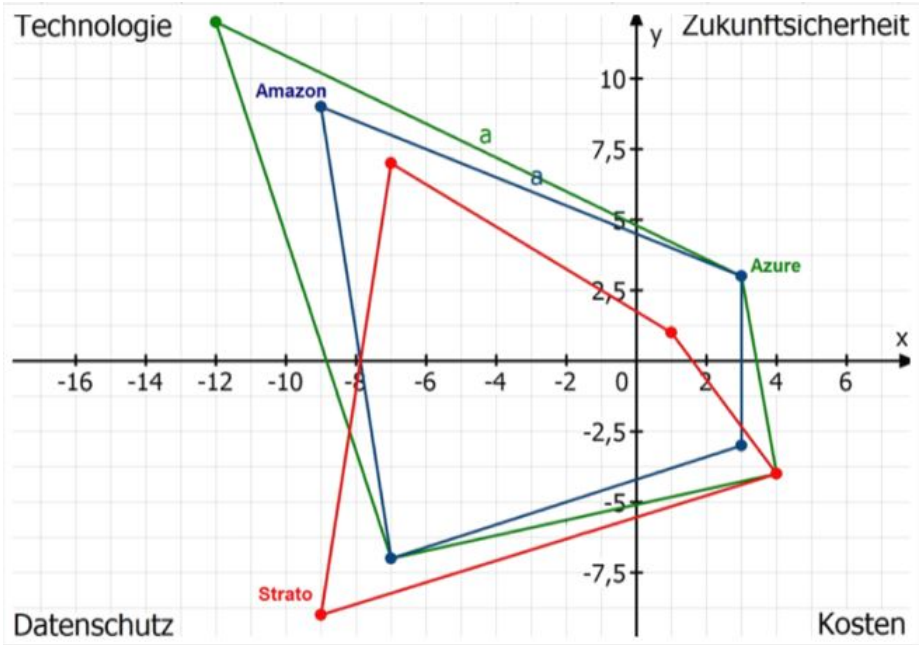


Abbildung A.11: Matrixdarstellung der Kosten von Azure, Amazon und Strato

Literaturverzeichnis

[1] *Berufsordnung für die Ärztinnen und Ärzte in Hessen.* – [Online] Verfügbar unter: http://www.laekh.de/upload/Rechtsquellen/berufsordnung.pdf. [Abgerufen am: 2015, Jul.24]

[2] *Bürgerliches Gesetzbuch.* – [Online] Verfügbar unter: http://www.gesetze-im-internet.de/bgb/. [Abgerufen am: 2015, Jul.24]

[3] *Bundesdatenschutzgesetz.* – [Online] Verfügbar unter: http://www.gesetze-im-internet.de/bdsg_1990/. [Abgerufen am: 2015, Jul.24]

[4] *Röntgenverordnung.* – [Online] Verfügbar unter: http://www.gesetze-im-internet.de/r_v_1987. [Abgerufen am: 2015, Jul.24]

[5] *Strafgesetzbuch.* – [Online] Verfügbar unter: http://www.gesetze-im-internet.de/stgb/. [Abgerufen am: 2015, Jul.24]

[6] *WhatsApp.* – [Online] Verfügbar unter: http://www.whatsapp.com. [Abgerufen am: 2015, Aug.03]

[7] Wilko Hartz: *Basiswissen QR-Code.* – [Online] Verfügbar unter: http://qrcode.wilkohartz.de. [Abgerufen am: 2015, Jun.25]

[8] Amazon: *Amazon - Web Services.* – [Online] Verfügbar unter: https://aws.amazon.com/de/. [Abgerufen am: 2015, Jul.03]

[9] Amazon: *Amazon EC2 - Preise.* – [Online] Verfügbar unter: https://aws.amazon.com/de/ec2/pricing/. [Abgerufen am: 2015, Jul.03]

[10] Amazon: *EU Datenschutz Whitpaper - Oktober 2014.* – [Online] Verfügbar unter: http://d0.awsstatic.com/whitepapers/compliance/De_Whitepapers/AWS_EU_Data_Protection_Whitepaper_DE_October_2014.pdf. [Abgerufen am: 2015, Jul.23]

[11] Andreas Dobler ; Rainer Kasan : *HealthDataSpace - Medizindaten Einfach Sicher.* – [Online] Verfügbar unter: https://my.healthdataspace.org. [Abgerufen am: 2015, Jun.17]

[12] Benjamin O. Orndorff: *Microsoft Azure.* – [Online] Verfügbar unter: https://azure.microsoft.com/de-de/. [Abgerufen am: 2015, Jul.03]

[13] Benjamin O. Orndorff: *Microsoft Azure - Preise der Datenübertragung.* – [Online] Verfügbar unter: http://azure.microsoft.com/de-de/pricing/details/data-transfers/. [Abgerufen am: 2015, Jul.03]

[14] Benjamin O. Orndorff: *Microsoft Azure - Preise des Cloud-Services.* – [Online] Verfügbar unter: https://azure.microsoft.com/de-de/pricing/details/cloud-services/. [Abgerufen am: 2015, Jul.03]

[15] BSI: *Grundlagen Cloud Computing.* – [Online] Verfügbar unter: https://www.bsi.bund.de/DE/Themen/CloudComputing/Grundlagen/Grundlagen_node.html. [Abgerufen am: 2015, Jul.09]

[16] Bundesärztekammer: *Empfehlung zur ärztlichen Schweigepflicht, Datenschutz und Datenverarbeitung in der Arztpraxis.* – [Buch] Deutsches Ärzteblatt, Jg. 111, Heft 21, 23.Mai 2014

[17] DENSO WAVE INCORPORATED : *QR Code - Answers to your questions about the QR Code.* – [Online] Verfügbar unter: http://www.qrcode.com/en/. [Abgerufen am: 2015, Jun.06]

[18] Dirk Paulus ; Christoph Ziegler ; Jochen Stange: *Medienagenten - Consulting, Design, Programmierung und PR.* – [Online] Verfügbar unter: http://www.medienagenten.de. [Abgerufen am: 2015, Jun.12]

[19] Dirk Paulus ; Christoph Ziegler ; Jochen Stange: *Vinopass - Mobile Weinvermarktung mittels QR-Codes.* – [Online] Verfügbar unter: http://vinopass.de/qr/. [Abgerufen am: 2015, Jun.12]

[20] Dr. Christian Böing: *Strato.* – [Online] Verfügbar unter: https://www.strato.de. [Abgerufen am: 2015, Jul.03]

[21] Dr. Christian Böing: *Strato - Server Cloud.* – [Online] Verfügbar unter: http://www.strato.de/server-cloud. [Abgerufen am: 2015, Jul.03]

[22] Dr. Friedrich Schwandt ; Tim Kröger: *Anzahl der Smartphone-Nutzer in Deutschland in den Jahren 2009 bis 2015 (in Millionen).* – [Online] Verfügbar unter: http://de.statista.com/statistik/daten/studie/198959/umfrage/anzahl-der-smartphonenutzer-in-deutschland-seit-2010/. [Abgerufen am: 2015, Jun. 02]

[23] Dr. Jürgen Serafin: *SIDEXIS 4.* – [Online] Verfügbar unter: http://www.sirona.com/de/produkte/bildgebende-systeme/sidexis-4/. [Abgerufen am: 2015, Jun.11]

[24] Enriquie Zabala: *Cloud Planet - Rijndael Cipher.* – [Online] Verfügbar unter: http://www.codeplanet.eu/files/flash/Rijndael_Animation_v4_eng.swf. [Abgerufen am: 2015, Jun. 27]

[25] Europäisches Parlament: *I. Richtlinie 95/46/EG des Europäischen Parlaments und des Rates vom 24. Oktober 1995 zum Schutz natürlicher Personen bei der Verarbeitung personenbezogener Daten und zum freien Datenverkehr.* – [Online] Verfügbar unter: https://www.datenschutzzentrum.de/material/recht/eu-datenschutzrichlinie.htm. [Abgerufen am: 2015, Jul.08]

[26] Frank Roselieb: *Der Online-GAU der Strato Medien AG - Ein Lehrstück für unzureichende Krisenkommunikation im Internet.* – [Online] Verfügbar unter: https://www.krisennavigator.de/Ereignisse-um-die-Strato-Medien-AG-im-Herbst-1999.113.0.html. [Abgerufen am: 2015, Jul.03]

[27] Gerald Münzl ; Michael Pauly ; Martin Reti: *Cloud Computing als neue Herausforderung für Management und IT.* – [Buch] ISBN: 978-3-662-45831-0, Heidelberg 2015

[28] Gerhard Lienemann: *Virtuelle Private Netzwerke. Aufbau und Nutzen.* – [Buch] ISBN 3-8007-2638-6, Vde-Verlag, Berlin u. a. 2002

[29] Hella Bissantz: *Zum QR-Projekt Kuns im öffentlichen Raum Frankfurt.* – [Online] Verfügbar unter: http://www.kunst-im-oeffentlichen-raum-frankfurt.de/de/page15.html. [Abgerufen am: 2015, Jun.12]

[30] Holger Bleich: *Hochverfügbare Ausfälle.* – [Online] Verfügbar unter: http://www.heise.de/ct/artikel/Hochverfuegbare-Ausfaelle-287780.html. [Abgerufen am: 2015, Jul.23]

[31] Jan Erlinghagen : *Experton-Studie: Der Cloud-Markt steht erst am Anfang.* 24. Juni 2013, . – [Online] Verfügbar unter: http://www.business-cloud.de/experton-studie-der-cloud-markt-steht-erst-am-anfang/. [Abgerufen am: 2015, Jun.10]

[32] Jinesh Varia ; Sajee Mathew: *Overview of Amazon Web Services - January 2014.* – [Online] Verfügbar unter: http://d36cz9buwru1tt.cloudfront.net/AWS_Overview.pdf. [Abgerufen am: 2015, Jul.019]

[33] Jon Galloway ; Christopher Harrison: *Introduction to ASP.NET MVC.* – [Online] Verfügbar unter: https://www.microsoftvirtualacademy.com/en-US/training-courses/introduction-to-asp-net-mvc-8322. [Abgerufen am: 2015, Jul.26]

[34] Kevin Remde: *SaaS, PaaS, and IaaS.* – [Online] Verfügbar unter: http://blogs.technet.com/b/kevinremde/archive/2011/04/03/saas-paas-and-iaas-oh-my-quot-cloudy-april-quot-part-3.aspx. [Abgerufen am: 2015, Jun.13]

[35] Klaus Schmeh: *Kryptografie - Verfahren, Protokolle, Infrastrukturen.* – [Buch] ISBN: 978-3-86491-267-2 5, aktualisierte Auflage Februar 2013

[36] Markus Repges: *AES (Rijndael).* – [Online] Verfügbar unter: http://www.repges.net/AES-Kandidaten/AES/aes.htm. [Abgerufen am: 2015, Jul. 27]

[37] Matthias Gärtner: *ISO 27001 Zertifizierung auf Basis von IT-Grundschutz.* – [Online] Verfügbar unter: https://www.bsi.bund.de/DE/Themen/ZertifizierungundAnerkennung/Zertifizierung-27001/GS_Zertifizierung_node.html. [Abgerufen am: 2015, Jul.29]

[38] Matthias Lohrer: *Einstieg in ASP.NET.* – [Buch] [Online] Verfügbar unter: http://openbook.rheinwerk-verlag.de/asp/index.htm. [Abgerufen am: 2015, Jul.09]

[39] Michael Canadi: *QR Codes - Einsatzmöglichkeiten in Mittelstand und Handwerk.* – [Online] Verfügbar unter: http://www.ebusiness-lotse-berlin.de/data/files/uin/Leitfaden_QR_Codes.pdf. [Abgerufen am: 2015, Jun.08]

[40] National Institute of Standards and Technology : *Federal Information Processing Standards Publication 197 - Announcing the Advanced Encryption Standard (AES).* – [Online] Verfügbar unter: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf. [Abgerufen am: 2015, Jun. 27]

[41] Nina Kurth: *Innovationen und ihre Bedeutung für Unternehmen.* – [Online] Verfügbar unter: http://hhc-duesseldorf.de/innovationen-und-ihre-bedeutung-fuer-unternehmen/. [Abgerufen am: 2015, Jun.29]

[42] Paul Echen: *Warum Datenschutz auch in kleinen Unternehmen wichtig ist.* – [Online] Verfügbar unter: http://www.lexware.de/unternehmer-und-organisation/warum-datenschutz-auch-in-kleinen-unternehmen-wichtig-ist. [Abgerufen am: 2015, Jul.23]

[43] Peter Kiesberg, Manuel Leithner, Martin Mulazzani, Lindsay Munroe, Sebastian Schittwieser, Mayank Sinha, Edgar Weippl : *QR Code Security.* – [Online] Verfügbar unter: http://www.sba-research.org/wp-content/uploads/publications/QR_Code_Security.pdf. [Abgerufen am: 2015, Jun.08]

[44] Planmeca OY : *Planmeca Romexis - Software für alle Bildgebungen.* – [Online] Verfügbar unter: http://www.planmeca.com/de/Dental-Software/. [Abgerufen am: 2015, Jun.17]

[45] Reiner Stauss: *Herausgabe von Behandlungsunterlagen – Pflichten des Zahnarztes und Abrechnungsmöglichkeiten.* – [Online] Verfügbar unter: http://www.iww.de/aaz/archiv/recht-herausgabe-von-behandlungsunterlagen–pflichten-des-zahnarztes-und-abrechnungsmoeglichkeiten-f29601. [Abgerufen am: 2015, Jun. 22]

[46] Sirona Dental Systems GmbH: *SIDEXIS 4 - Handbuch für den Anwender.* – [Buch] Bestellnr.: 64 47 010 D3592, April 2014

[47] Srdan Dzombeta ; Oliver Goldstein: *Whitepaper - Patientendaten und Cloud Computing.* – [Online] Verfügbar unter: http://www.persicon.com/news/items/patientendaten-und-cloud-computing.html. [Abgerufen am: 2015, Jul.03]

[48] Statista GmbH: *Nettoumsatz von Amazon.com in den Jahren 2004 bis 2014 (in Milliarden US-Dollar).* – [Online] Verfügbar unter: http://de.statista.com/statistik/daten/studie/75292/umfrage/nettoumsatz-von-amazoncom-seit-2004/. [Abgerufen am: 2015, Jul.22]

[49] Statista GmbH: *Umsatz der Microsoft Corporation in den Geschäftsjahren 2002 bis 2015 (in Milliarden US-Dollar).* – [Online] Verfügbar unter: http://de.statista.com/statistik/daten/studie/155707/umfrage/entwicklung-des-umsatzes-der-microsoft-corporation-seit-dem-geschaeftsjahr-2002/. [Abgerufen am: 2015, Jul.20]

[50] TÜV SÜD: *ISO 27001 - InformationInformation durch ISO/IEC 27001.* – [Online] Verfügbar unter: http://www.tuev-sued.de/management-systeme/it-dienstleistungen/iso-27001. [Abgerufen am: 2015, Jul.29]

[51] Ulrike Wintergalen: *Erneuter Strato Verkauf - Die Strato Geschichte.* – [Online] Verfügbar unter: https://www.webwork-magazin.net/strato-verkauf-die-strato-geschichte/627. [Abgerufen am: 2015, Jul.23]